MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

12

Cr$ 8.200

Andróides e autômatos • Menus
Raio X da câmara eletrônica

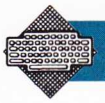
rioGráfica

MANAUS, ALTAMIRA, MACAPÁ, JI.PARANÁ, PORTO VELHO, RIO BRANCO, SANTARÉM, BOA VISTA, SINOP, ALTA FLORESTA: Cr$ 10.660

MICROCOMPUTADOR CURSO PRÁTICO

APLICAÇÕES

Homens e máquinas 237



PERIFÉRICOS

Desempenho versátil 240

PROJETOS DE PROGRAMAS

Um novo ângulo 242

OFICINA

Circuitos e componentes 244

RAIO X

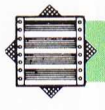
Câmara escura 246

SOFTWARE

A produção de softwares 248

LINGUAGENS

Em conjunto 251



TÉCNICAS DE PROGRAMAÇÃO

Faça sua escolha 254

PERFIL

Radiocontato 256

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1) Qual o nome da ciência que estuda os sistemas de comunicação e controle dos seres vivos e das máquinas, e que foi fundamental para o desenvolvimento da robótica?
2) Qual o principal cuidado que se deve tomar antes de fazer qualquer reparo num micro?
3) Qual a linguagem mais usada atualmente pelas grandes software houses internacionais para o desenvolvimento de programas?
4) Qual o microprocessador de 32 bits, utilizado pelos computadores Lisa e Macintosh?

RESPOSTAS DO TESTE ANTERIOR

1) A fita do Wafadrive notabiliza-se por ser do tipo usado para vídeo, e correr num anel sem fim.
2) Um conjunto de softwares independentes mas compatíveis entre si, interligados pelo sistema operacional, é uma solução que integra várias aplicações com economia de memória.
3) Os comandos PASCAL que controlam os loops condicionais são REPEAT e WHILE.
4) É o comando TRUNC em PASCAL que remove a fração decimal de qualquer número.

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as 3.as e 4.as trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 30 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem as 3.as e 4.as capas do fascículo 30); 3.as e 4.as capas dos fascículos de 1 a 29; e guardas.

rioGráfica **Diretoria Executiva:** Oscar Neves; Filipe Zander; Nilo Sérgio de Almeida; Danilo Esteves Costa; **GRUPO EDUCAÇÃO E CULTURA - Planejamento e Comercialização - Gerente de Planejamento:** João Azeredo Coutinho; **Gerente de Marketing:** Jaime Rodrigues; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Ayrton Menezes; **Diretor de Vendas:** Marcos March; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto; **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Mauricio Rittner; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Revisores:** Isis Augusta Loyolla; Cacilda Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Adalberto Cornavaca; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistente de Arte:** Miguel Luis Escámez Simón; **Colaboradores - Editores-assistentes:** Claudio Marcondes; Isa Mara Lando; Luiz Carlos Pizarro Marin; Nadya Morelo Reis; **Texto:** Paulo Brito; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Fernando Dorfman Knijnik; Lúcia Maria Leal Gonçalves; Maria Clara Cescato; Maria Ester Menezes Martinho; Newton Roberval Eichemberg; Patricia de Paiva e Castro; Regina Amarante; **Pesquisa:** Stella Maria Quentel; **Arte:** Edemilson Ribeiro da Silva; **Assessoria Técnica:** Geraldo Coen; Gustavo José Ferreira Grubba; Lauro de Lauro; Pierluigi Piazzi; Sérgio Jorge Batarce; Imarés Microcomputadores. As fotos não creditadas pertencem à obra original. Seguem-se os créditos das demais: Fernando Scavone; Jorge's Estúdio. © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** Editora Rio Gráfica Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG.

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Ltda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

HOMENS E MÁQUINAS

Máquinas que imitam o homem pela forma ou ação, os robôs constituem um tema sempre fascinante. Daí sua constante evolução, dos bonecos mecânicos do século XVIII aos robôs industriais de hoje.

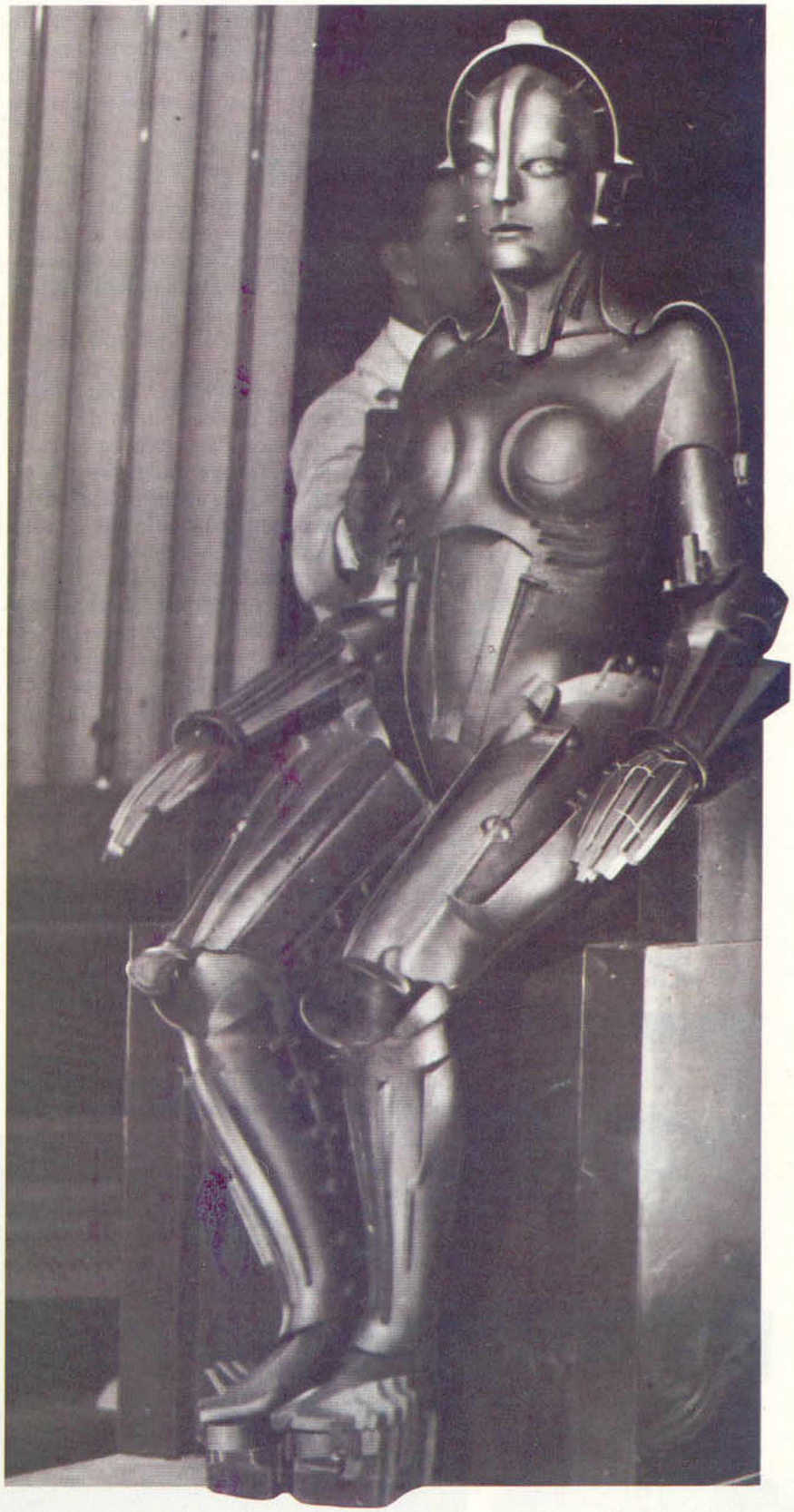
Há séculos que as pessoas são atraídas pela idéia da existência de homens mecânicos. Filósofos, engenheiros, inventores, em várias épocas, pensaram na criação de mecanismos que imitassem o comportamento humano. Embora os robôs modernos tenham a aparência menos humana e sejam projetados para desempenhar uma série de ações específicas, geralmente nas indústrias, os bonecos mecânicos do passado eram projetados para ter a aparência mais viva possível e dar a impressão de que seriam capazes de praticar qualquer ação humana.

No entanto, o primeiro robô não se parecia com um homem: parecia um pato. Em 1738, Jacques de Vaucanson (1709-1782), um engenheiro francês, apresentou um pato mecânico à Academia Real de Ciências, em Paris. Seu invento batia as asas, grasnava e comia. Mais tarde, ainda no século XVIII, um inventor suíço, Pierre Jacquet-Droz (1721-1790), criou um conjunto de bonecos mecânicos capazes de variadas ações. Um escrevia, outro desenhava, outro tocava órgão. No final do século XIX já havia grande número desses brinquedos, todos baseados em mecanismos de relojoaria.

Os primeiros andróides

Nesse período, construíram-se numerosas figuras com aparência humana extraordinariamente verossímil, cujos movimentos nem sempre eram baseados em peças de relógio. Em 1893, George Moore fez um homem mecânico, movido a vapor, que produzia um efeito interessante: o vapor que saía de sua boca parecia a fumaça de um charuto que ele "fumava".

Tecnologias mais evoluídas, contudo, estimularam o processo de aperfeiçoamento dessas máquinas: desde simples brinquedos mecânicos capazes de andar até o clássico Elektro, andróide construído pela empresa americana Westinghouse. Com 2,15 m de altura, era capaz de dizer oitenta palavras, contar, andar, cumprimentar e reconhecer cores. Movido por onze motores elétricos, pesava 117 kg. Para controlar essa imensa figura, a Westinghouse desenvolveu um "cérebro" composto por 82 relés.



Mas cada um desses homens mecânicos tinha suas limitações. Além do valor como entretenimento, nenhum deles era hábil o bastante para ser considerado um robô ideal. Um homem mecânico que sabe desenhar não pode fazer as compras por você. E, mesmo sabendo andar por uma sala, seria incapaz de chegar a uma loja sem an-

O antepassado
Este robô foi o primeiro a aparecer no cinema: está no filme de Fritz Lang *Metropolis* (1926), que, em 1984, teve uma versão sonorizada e parcialmente colorida por computador.

tes se chocar com um poste. Todos eram máquinas, simplesmente máquinas — apenas realizavam uma série de ações que não requeriam tomada de decisões e não demonstravam nenhum sinal de inteligência.

Robôs na ficção

Mas, se os inventores e engenheiros esbarravam sempre em limitações, os escritores de ficção científica superavam todas com o uso da imaginação. Esse gênero de literatura desenvolveu-se bastante com a idéia do robô.

A própria palavra "robô" é produto de uma obra de ficção: foi criada em 1923 pelo autor tcheco Karel Capek (1890-1938). Na peça *Robôs universais de Rossum,* Capek tratava da invenção de homens mecânicos tão perfeitos que executavam todas as tarefas realizadas por um ser humano. No final, os robôs achavam que os humanos não serviam para nada, colocando-os em uma situação dramática. Em tcheco, a palavra *robota* significa "trabalhador". Assim, o título da peça deveria ser traduzido por "Trabalhadores universais de Rossum", mas a palavra "robô" surtiu inesperado efeito, transformando-se no termo padrão para qualquer ser mecânico com habilidades humanas (ou semelhantes).

As fantasias da ficção sobre criaturas construídas à imagem do homem encontram-se também no romance *Frankenstein* (1818), de Mary Shelley. Embora não fosse mecânico, o monstro criado por Victor Frankenstein era constituído por um conjunto de partes obtidas de outros seres humanos. As criaturas invasoras de *Guerra dos mundos* (1898), de H. G. Wells, eram robotizadas, ainda que parcialmente.

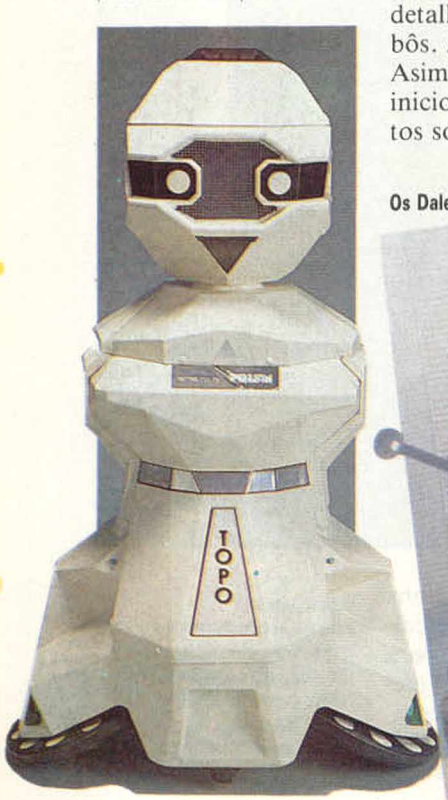
Os escritores do século XX têm explorado em detalhes um mundo imaginário habitado por robôs. A contribuição mais notável foi a de Isaac Asimov, famoso escritor de ficção científica, que iniciou sua carreira em 1940 escrevendo contos sobre robôs e seus problemas operacionais. O mundo dos robôs de Asimov é tão bem estruturado que ele chegou a formular as três "leis da robótica" no "Manual de robótica (56.ª edição, 2058 d.C.)", do romance *Eu, robô.*

Os robôs também assinalam sua presença no cinema e na televisão. Por exemplo, nas séries de tevê sobre ciborgs *(O homem de 6 milhões de dólares* e *A mulher biônica)* e no filme *Guerra nas estrelas,* em que C3PO e R2D2 apresentam-se em igualdade com os astros humanos com quem contracenam.

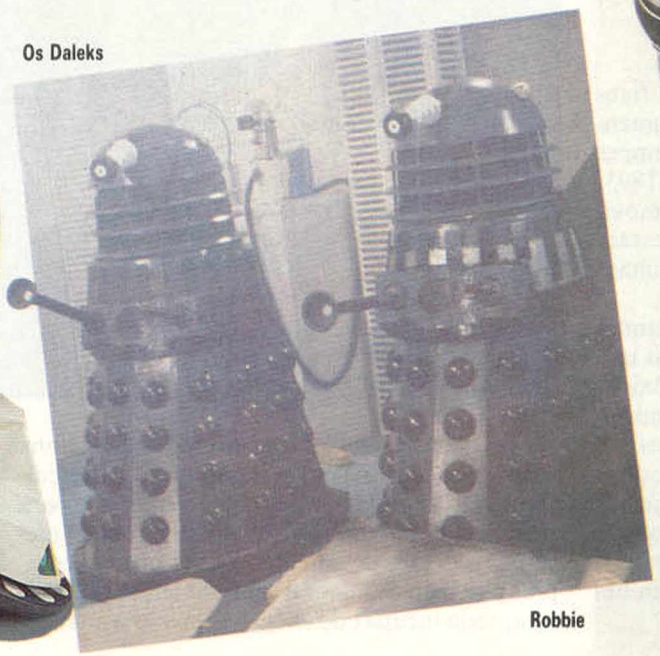
Comparado a tais arrojos de fantasia, o emprego real dessas máquinas parece banalidade. Mas são os robôs industriais, encontrados nas linhas de montagem, que recebem a atenção dos homens de negócios: as estimativas para 1985 eram de 25.000 robôs em uso na indústria japonesa, 15.000 nos Estados Unidos e 8.000 na Alemanha Ocidental. No Brasil, eles começaram a ser usados na indústria automobilística, em 1982.

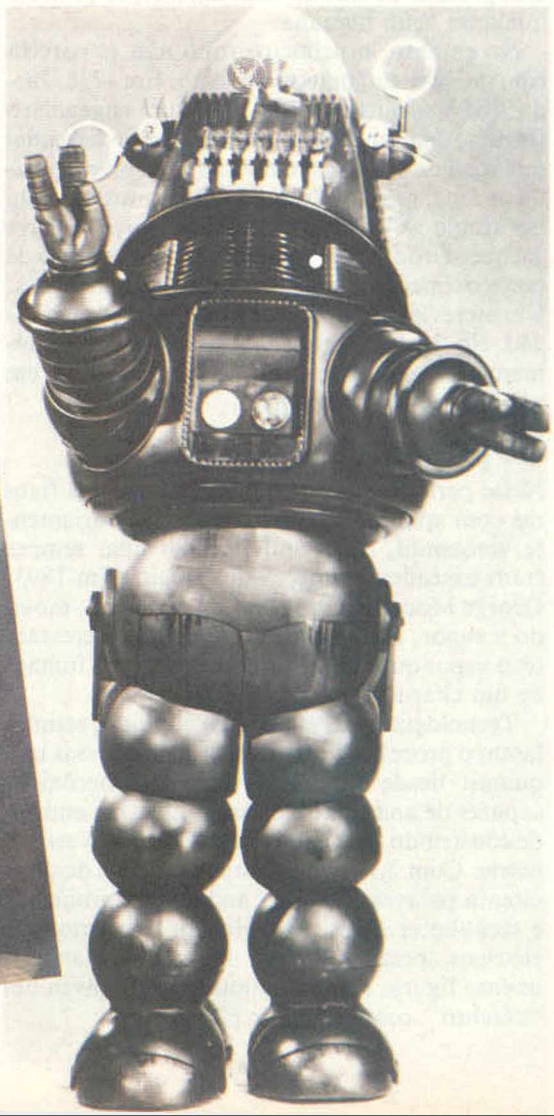
Para a maioria das pessoas, porém, os robôs industriais são desinteressantes. Uma máquina que passa o tempo todo soldando chassis de automóvel ou pintando carroçarias está longe da imagem criada pela ficção. Se o robô ideal virá um dia a ser desenvolvido ou não, é questão para discutir. E dificilmente se pode determinar se ele será projetado à imagem do homem.

Ficção e realidade
Os Daleks, certamente os robôs mais famosos do mundo, durante muitos anos foram heróis de um seriado de tevê europeu. Mas Robbie, de *Planeta proibido,* personificou a fantasia de um robô que tem até emoções. Todos muito diferentes de Topo *(abaixo),* robô de verdade para uso doméstico, produzido na Europa.



Os Daleks





Robbie

Os falsos homens

Pelo menos na ficção, os robôs receberam tantos nomes que se tornou necessário um glossário com os termos mais usados.

Andróide: Robô projetado de modo a parecer um ser humano, em todos os aspectos.

Antropomorfo: Significa, literalmente, "com forma humana". Assim, todo andróide é antropomorfo; mas muitos robôs só têm algumas partes com essa característica. Por exemplo, podem ter apenas braços antropomórficos.

Automação: Controle automático de qualquer processo de fabricação.

Autômato: Máquina com mecanismos ocultos que realiza uma série de operações predeterminadas. Os primeiros homens mecânicos eram autômatos. Mas a palavra tem significado mais profundo quando aplicada à teoria dos autômatos, um sistema analítico por meio do qual um dispositivo qualquer (um robô, um computador ou mesmo uma pessoa) pode ser estudado e descrito.

Cibernética: Estudo dos sistemas de controle e comunicações, tanto dos seres vivos como das máquinas. Criada por Norbert Wiener em 1947, baseia-se na possibilidade de ser usada para examinar sistemas biológicos como se estes fossem máquinas.

Cibert: Humanóide mecânico.

Ciborg: Do inglês cybernetic organism (organismo cibernético). Organismo no qual há partes mecânicas e partes biológicas.

Cibot: Robô que raciocina.

Colarinho metálico: Robôs industriais. Quem trabalha em escritório é designado como "colarinho branco"; se trabalha em produção, como "colarinho azul". Por analogia, o robô acabou sendo o "colarinho metálico".

Dróide: Robô que obedece às leis da robótica, de Asimov.

End effector: Termo técnico inglês usado para designar a mão de um robô.

Homúnculo: Homem pequenino ou boneco.

Manipulador: Mão de robô.

Robô: Máquina capaz de executar algumas funções humanas, embora não precise ter semelhança física com o homem.

Robótica: Ciência que estuda os robôs.

As leis da robótica

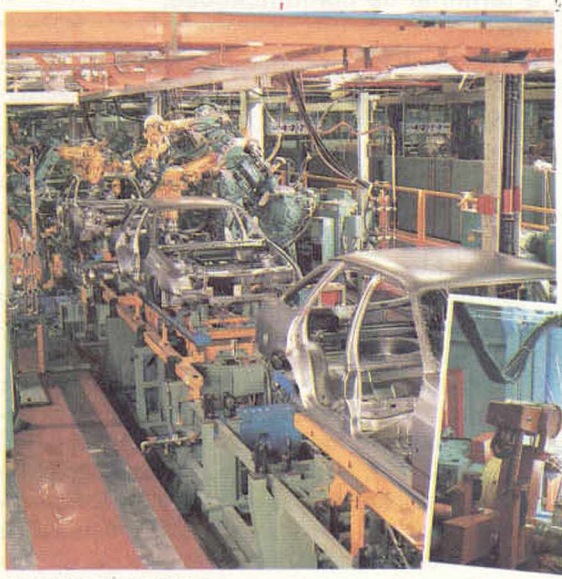
1. Um robô não pode ferir um ser humano nem permitir, por omissão, que um ser humano seja ferido.

2. Um robô deve obedecer às ordens dadas pelos seres humanos, exceto quando conflitarem com a primeira lei.

3. Um robô deve proteger sua existência enquanto isso não conflitar com as leis anteriores.

Do livro *Eu, robô*, de Isaac Asimov.

As regras
Quando os robôs forem capazes de distinguir um ser humano de uma pedra, as leis de Asimov *(acima)* poderão perfeitamente formar a diretriz de seu comportamento.



Ford: linha de montagem

Fiat: soldagem de componentes

Linha de montagem
Atualmente, os robôs são empregados sobretudo na indústria. Fazem o trabalho com mais rapidez e precisão que o homem e constituem ferramentas importantes em ambientes insalubres como as seções de soldagem e pintura das indústrias automobilísticas. Em geral, o robô todo é apenas um ou dois braços mecânicos.

O que é um robô?

Há dezenas de definições, mas no mundo inteiro o termo "robô" se tornou genérico para designar todas as máquinas que têm alguma semelhança com o homem e, também, máquinas industriais "inteligentes", dotadas de movimento e que podem ser programadas para inúmeras tarefas, como soldar, apertar porcas ou parafusos, pintar.

Teoricamente, um robô pode fazer tudo; as limitações ficam por conta de quem o projeta e da tecnologia disponível. Além de se movimentarem, os robôs "vêem", "ouvem", "falam". Certos dicionários já os definiram como máquinas capazes de executar algumas funções humanas (embora sem necessidade de se assemelharem ao homem), mas essa definição perdeu o sentido.

Sua forma e capacidade para desempenhar tarefas dependem apenas do que pretendemos e do que conseguimos que ele faça. Assim, pode ser que um robô de solda empregado numa indústria não se mova porque foi projetado para ficar apenas num lugar. Já um robô doméstico, embora capaz de fazer café, talvez não consiga subir as escadas e levar a bandeja até o quarto de seu dono.

DESEMPENHO VERSÁTIL

Embora tenha a aparência de simples máquina de escrever elétrica, a Brother EP-44 é muito versátil, e funciona como impressora, calculadora e terminal de comunicações.

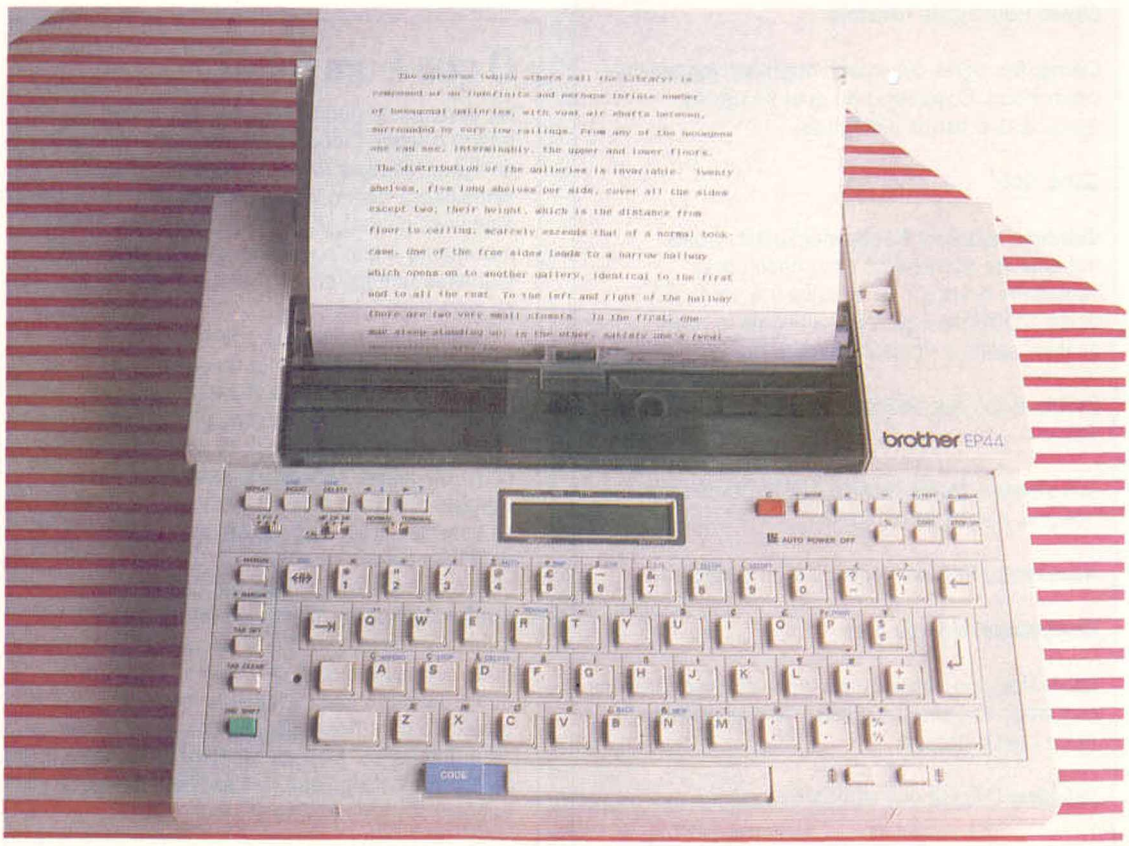
A Brother EP-44 pesa pouco mais de 2 kg e é alimentada a pilhas, o que a torna uma verdadeira máquina portátil. Usando transformador, pode ser acionada por corrente elétrica. Além das teclas convencionais de máquina de escrever, a Brother EP-44 tem quatro outras para o processamento de texto e sete para a função de calculadora. Mas a característica mais notável é o visor de cristal líquido (LCD). Esse dispositivo possibilita ao usuário visualizar e corrigir o texto na tela antes de imprimi-lo.

Para usar a EP-44 como máquina de escrever comum, basta colocar o Print Mode Selector (seletor de modo de impressão) em Direct Print (impressão direta). Nesse modo, o texto aparece no LCD. Margeamento, tabulação e espacejamento de linhas são fixados como nas máquinas de escrever convencionais. No entanto, há um sistema incomum de alimentação de papel — em vez de girar manualmente o rolo, o usuário pressiona um botão a fim de mover o papel para cima ou para baixo.

A EP-44 utiliza uma cabeça de impressão térmica para queimar uma série de pontos no papel e assim formar o caractere desejado. A maioria das impressoras térmicas tem matriz de 9 x 7 pontos para formar os caracteres, mas a matriz de 24 x 18 da EP-44 fornece melhor qualidade de impressão.

A maior desvantagem do sistema térmico de impressão está na baixa velocidade da impressão — nesse caso, menos de dezesseis caracteres por segundo. Quando a Brother é usada como máquina de escrever, isso não chega a ser um inconveniente, mas, como impressora acoplada a um micro, a diferença é considerável: uma boa impressora matricial é dez vezes mais veloz que a Brother. O outro problema com esse sistema está na necessidade do papel térmico especial, muito caro.

Se o seletor de modo de impressão é posicionado em Correction Print (impressão com correção), o LCD mostra toda a sua utilidade. Nesse modo, o texto aparece no visor e há um intervalo antes da impressão: o texto registrado no papel fica quinze caracteres atrás da letra que está



O teclado
Há duas teclas para maiúsculas com trava, uma tecla para símbolos e outra para função. As teclas de caracteres são multifuncionais, produzindo acentos e uma série de caracteres em línguas estrangeiras. Teclas de função controlam os modos de impressão, de cálculo e de memória, o ajuste das margens, as tabulações e o movimento do papel.

sendo digitada. Esse recurso permite que você altere qualquer um dos quinze últimos caracteres antes que sejam impressos — o que será útil se sua datilografia não for muito boa e constitui grande vantagem com relação aos líquidos corretores. A correção é feita no visor pelo uso das duas teclas de movimentação do cursor para escolher o caractere a ser alterado.

Há também o modo Line-by-Line (linha por linha), no qual a EP-44 espera até que a tecla [Return] seja pressionada, antes de imprimir uma linha inteira. Isso dá a chance de se corrigir toda uma linha de texto em vez de apenas os quinze últimos caracteres. Nesse modo, o LCD age como uma janela que se desloca ao longo da linha de texto por meio das teclas do cursor.

A EP-44 possui o modo Auto Return (retorno automático), no qual uma nova linha será iniciada se a barra de espaço for pressionada a seis toques (ou menos) do final da linha.

A calculadora incorporada na EP-44 possibilita a execução de contas no visor, sem prejudicar a impressão.

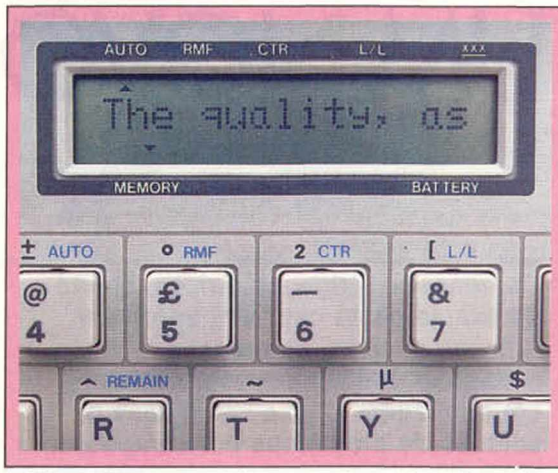
As diferenças

Todos esses recursos são encontrados em muitas das modernas máquinas de escrever elétricas. Mas a EP-44 pode também ser usada como uma processadora de texto simples. Para isso, dispõe de 3,5 Kbytes de memória interna para armazenar texto. Essa capacidade de memória parece pequena quando comparada à disponível na maioria dos micros, mas é suficiente para produzir uma carta de três páginas. Os comandos necessários para o processamento de texto são acessados pelo uso das teclas normais de máquina de escrever em conjunção com a tecla [Code]. Assim, para entrar com texto na memória, basta pressionar a [Code] e a letra N — que, no caso, significa "novo texto". A máquina aceita o texto tanto no modo direto como na impressão com correção.

Armazenado o texto na memória, usam-se o visor e as teclas do cursor para corrigi-lo. As teclas do cursor permitem mover o foco do LCD para cima, para baixo, para a esquerda e para a direita; assim, todo o texto pode ser visualizado e caracteres são inseridos, eliminados ou substituídos. A única restrição: não se deslocam palavras inteiras para a linha de baixo quando a linha que está sendo corrigida é muito longa — o que significa que você não pode reformatar toda a diagramação de um documento.

A grande vantagem de armazenar textos na memória está na possibilidade de modificá-los sem necessidade de redatilografar todo o documento. Se você quiser enviar a mesma carta a várias pessoas, precisará apenas modificar o texto na memória, inserindo o nome e o endereço desejados, e pressionar a tecla [Code] juntamente com a letra P (de Print, imprimir).

O texto permanece na memória da EP-44 mesmo depois de desligada, o que permite entrar com ele e corrigi-lo depois.



LCD

O visor de cristal líquido da EP-44 mostra quinze caracteres, cinco indicadores de modo e um aviso de descarga das pilhas. O texto pode ser editado na tela, permitindo processamento. A luminosidade dos caracteres no visor ajusta-se conforme as condições de iluminação do ambiente.

Uma chave marcada com a inscrição "Terminal" transforma a Brother em impressora de computador. Mas alguma experimentação é necessária para ajustar a velocidade de transmissão correta, o comprimento das palavras e outros aspectos relativos a seu micro.

É fácil compatibilizar a EP-44 com a interface serial do microcomputador: o visor mostra as várias velocidades de transmissão e outros parâmetros e você pressiona a chave Mode quando o parâmetro for adequado para sua máquina. Essa configuração mantém-se mesmo depois que se desliga a EP-44.

Versatilidade compactada

A Brother não pode lidar com formulários contínuos, e as folhas de papel devem ser colocadas manualmente, uma a uma. Mas a qualidade da impressão é muito melhor que a de uma impressora matricial; assim, a EP-44 é ideal para a produção de cartas e documentos curtos.

A Brother tanto envia como recebe dados. Conectada a um modem, permite que você se comunique por telefone com outros usuários, ou com máquinas maiores, que empregam padrões compatíveis de comunicação.

A Brother EP-44 destaca-se pela versatilidade e por reunir tantas funções diferentes em sua estrutura compacta. No mercado internacional, entra na avaliação de qualquer usuário de micro que esteja procurando uma impressora utilizável também como máquina de escrever.

Print-Out

Any smooth-finish typing paper can be used with the ribbon in place, but the paper-feed cannot accept carbons. The print quality is good on both types of paper.
Text can be
  Centred & Underlined
foreign (Æø¿¢µ¶), and sub- or super-scripted ($X^{2}=N_{2}+T_{0}$).

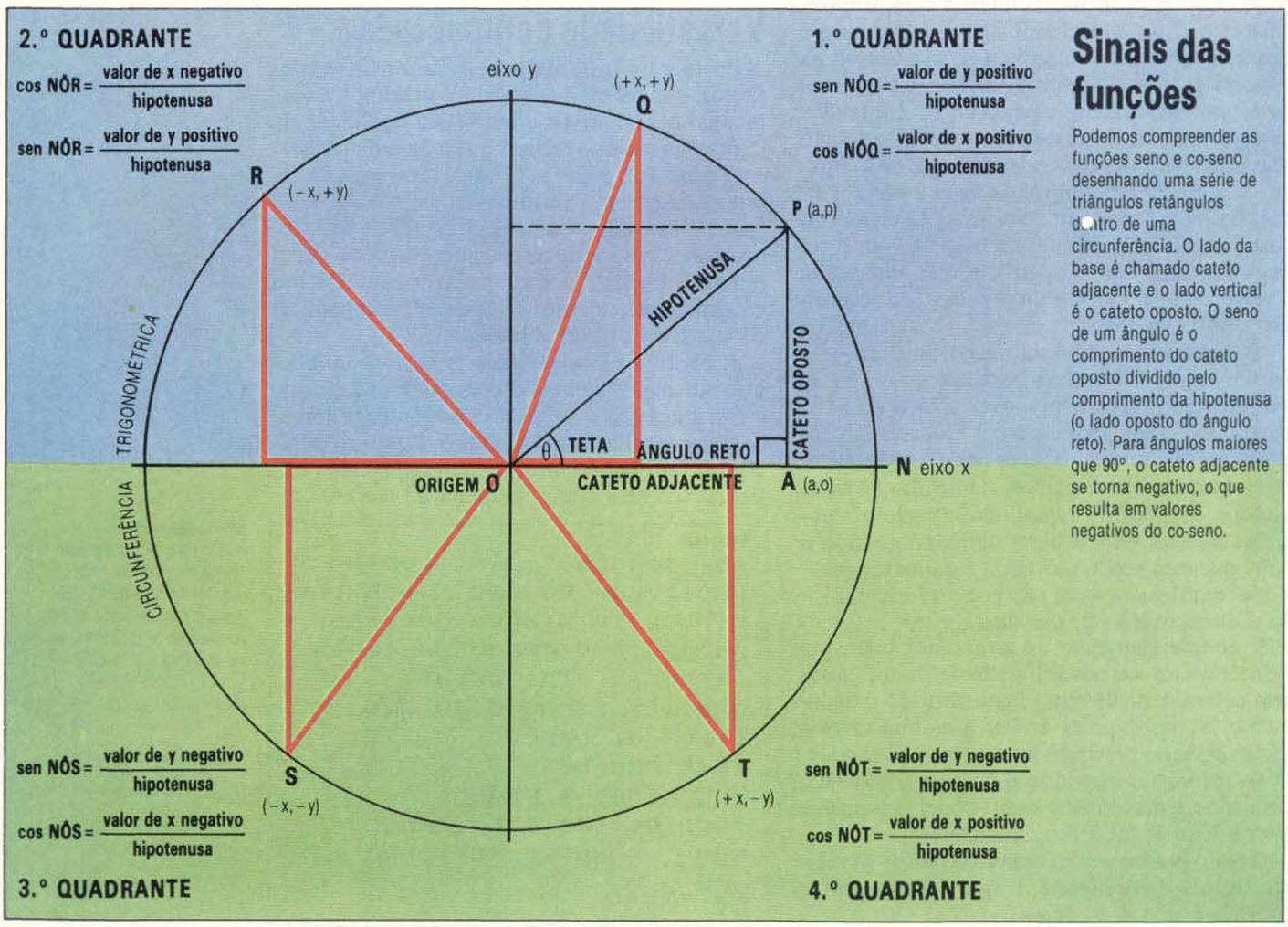
Melhor de dois
Nesta foto são mostrados os vários formatos de impressão oferecidos pela EP-44, que combina os recursos de um micro com os de uma máquina de escrever.

UM NOVO ÂNGULO

Os conceitos trigonométricos, objeto de exame deste capítulo, envolvem funções básicas, como seno e co-seno, facilmente codificadas em linguagem BASIC e vantajosas na solução de problemas por computador.

De que tipo de matemática os programadores realmente precisam? Isso depende do programa que estão desenvolvendo. As diferentes versões da linguagem BASIC incorporam os inúmeros comandos e funções para lidar com desenhos: PLOT, CIRCLE, FILL, LINE, COLOR, INK, PAPER etc. Assim, a rotação e a translação de figuras simples na tela não apresentam grandes problemas, uma vez que alguns comandos já incluem os cálculos matemáticos necessários à realização desses movimentos. Há ocasiões, no entanto, em que são necessárias funções trigonométricas como SIN (seno), COS (co-seno) e TAN (tangente), o que também não constitui dificuldade porque a linguagem BASIC possui um bom conjunto delas. Se esses termos da trigonometria nada significam para você, não se preocupe — todos serão logo explicados.

Os estudantes de matemática do 2.° grau costumam perguntar sobre a relevância prática da trigonometria. Sob certo aspecto, a trigonometria é considerada uma ligação entre a geometria euclidiana, que lida com pontos, linhas e curvas, e a álgebra, que possibilita chegar a soluções matemáticas pela manipulação de variáveis com valores desconhecidos. A parábola, uma figura geométrica plana, serve como exemplo para comparar as soluções algébricas e geométricas. Suas propriedades podem ser observadas traçando-se a curva com o uso de papel quadriculado, transferidor, régua e lápis. No entanto, muito mais útil é perceber que a curva



Sinais das funções

Podemos compreender as funções seno e co-seno desenhando uma série de triângulos retângulos dentro de uma circunferência. O lado da base é chamado cateto adjacente e o lado vertical é o cateto oposto. O seno de um ângulo é o comprimento do cateto oposto dividido pelo comprimento da hipotenusa (o lado oposto do ângulo reto). Para ângulos maiores que 90°, o cateto adjacente se torna negativo, o que resulta em valores negativos do co-seno.

pode ser representada pela fórmula algébrica $y = x^2$.

Essa fórmula permite calcular os valores para qualquer ponto da curva sem que se tenha de desenhá-la. Os problemas apresentados sob a forma algébrica prestam-se muito mais a soluções por computador do que os que requerem gráficos ou figuras; assim, torna-se evidente a vantagem das soluções trigonométricas.

O objetivo deste artigo é fazer uma rápida recapitulação dos conceitos trigonométricos, examinando suas funções básicas: seno e co-seno.

Razão entre lados

O seno e o co-seno de um ângulo ou de um arco representam a razão entre dois lados de um triângulo retângulo. Por se referirem a ângulos e arcos, essas funções estão diretamente relacionadas com a circunferência. Qualquer triângulo retângulo pode ser inscrito, ou seja, desenhado de forma que seus vértices estejam sobre a circunferência. A circunferência trigonométrica é também chamada unitária porque se atribui valor unitário a seu raio — a medida real não importa, são as proporções que interessam. A primeira ilustração mostra o segmento de reta ON, que descreve um ângulo θ, num movimento de rotação em torno do ponto O. A posição inicial de rotação, por convenção, é o eixo horizontal, no sentido anti-horário. O eixo horizontal é o eixo x ou eixo dos co-senos, e o ângulo de rotação costuma ser designado por letras do alfabeto grego. Traçando-se a partir de um ponto da circunferência uma linha perpendicular ao eixo x, obtém-se um triângulo retângulo.

O co-seno de θ é definido como a razão entre o cateto adjacente ao ângulo (lado OA, contido no eixo x) e a hipotenusa (OP, raio da circunferência unitária). Consideremos, por exemplo, um ângulo de 35° numa circunferência com raio de 15 mm: a medida do lado adjacente ao ângulo é de cerca de 12,28728 mm. Dividindo esse valor por 15, obtemos 0,819152, que é o valor do co-seno de 35°. Se você tiver uma calculadora científica com tecla [Cos] ou uma tabela de co-senos, tente entrar com [35 Cos]. Deverá obter o resultado 0,819152044. Essa razão é verdadeira para qualquer tamanho de circunferência unitária na qual o triângulo retângulo esteja inscrito. Tenha o raio do círculo 1 cm, 1 km ou um ano-luz, o lado do triângulo contido no eixo x sempre medirá cerca de 0,82 do valor do raio.

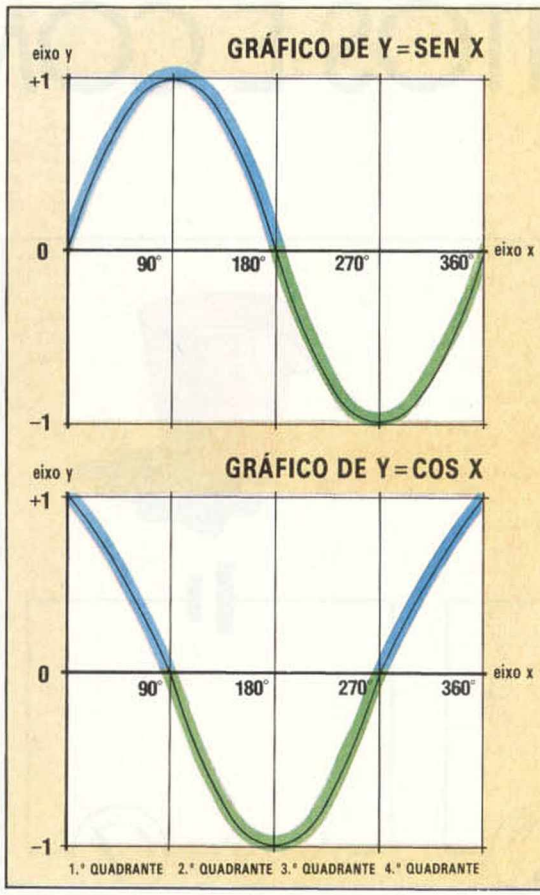
É possível também desenhar outros ângulos na circunferência unitária da ilustração. Pode-se ver que, para qualquer ângulo θ, o valor de cos θ (cos é a representação algébrica de co-seno) nunca será maior que 1 nem menor que 0. No entanto, para valores de θ maiores que 90°, o cos θ será negativo. Isso porque, à esquerda da origem (O), os valores das coordenadas x são negativos, e à direita, positivos. Assim, para ângulos θ maiores do que 90°, o cateto adjacente estará do lado negativo, o que implica valores negativos para cos θ. Pelo mesmo motivo, os ângulos entre 180° e 270° também terão co-senos negativos.

No entanto, entre 270° e 360° (o que corresponde a uma volta completa), o co-seno será novamente positivo.

Para maior facilidade de localização na circunferência trigonométrica, costuma-se dividi-la em quatro partes iguais — os quadrantes —, compreendendo cada uma ângulos de 90°. Assim, o 1.° quadrante abrange os ângulos de 0° a 90°; o 2.° vai de 90° a 180°; o 3.°, de 180° a 270°; e o 4.°, de 270° a 360°.

A função seno de um ângulo é muito semelhante ao co-seno, exceto que envolve valores no eixo y (o eixo vertical). Por uma simples translação, pode-se ver que os valores no eixo y coincidem com o comprimento do cateto oposto ao ângulo. Se θ for 0, o lado adjacente ao ângulo será igual ao comprimento da hipotenusa. A coordenada de P no eixo x será sempre 1 (porque 1/1 = 1), mas a coordenada do eixo y será 0 (ou seja, o cateto oposto será 0). Para todos os valores de 0° a 90°, sen θ (representação algébrica de seno) variará de 0 a 1. No segundo quadrante da circunferência, sen θ também será positivo, mas decrescerá de um máximo de 1 até 0, à medida que θ for se aproximando de 180°. Todos os ângulos θ maiores que 180° até (mas não inclusive) 360° terão senos negativos.

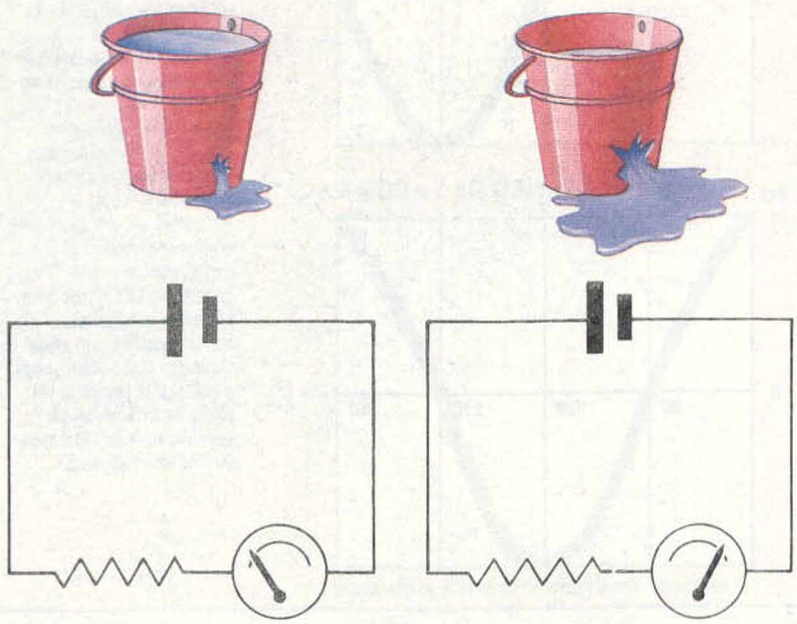
Outra função trigonométrica muito utilizada é a tangente, definida como a relação entre o seno e o co-seno, o que equivale à razão entre o cateto oposto e o adjacente ao ângulo.



Ondas

O desenho da senóide é produzido com os valores assumidos pela função seno numa circunferência completa. Observe que os eixos x e y aqui representados nada têm a ver com os eixos x e y que separam os quadrantes da circunferência trigonométrica. Ao longo do eixo x estão os ângulos de 0° a 360°, e no eixo y encontram-se os valores dos senos desses ângulos. Todos os valores do seno estão entre +1 e −1. As quatro seções do gráfico correspondem aos quadrantes; o seno é positivo (em azul) nos dois primeiros quadrantes e depois negativo (em verde). O gráfico do co-seno chega a resultados semelhantes: tem a mesma forma da senóide, apenas deslocado 90° ao longo do eixo x.

CIRCUITOS E COMPONENTES

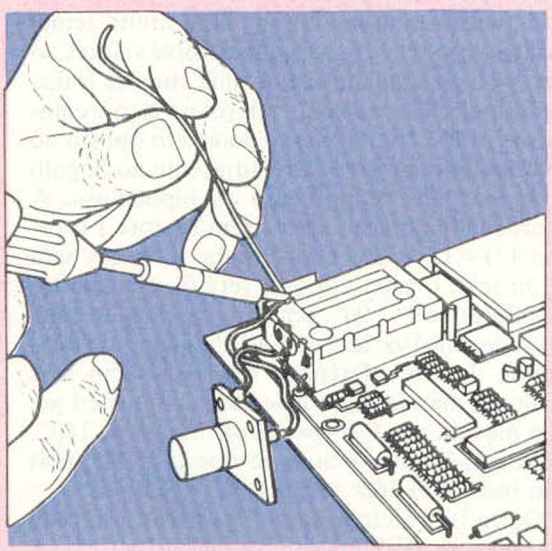


Resistência e corrente

Num circuito elétrico, todos os componentes opõem resistência ao fluxo de eletricidade, da mesma forma que, num sistema de aquecimento central, os canos e reservatórios resistem à passagem da água: quanto maior a resistência apresentada, menor o fluxo. De um balde com um pequeno furo pouca água escapa, porque as bordas do furo resistem ao fluxo; se o furo for maior, a resistência será menor e, em conseqüência, sairá mais água. Nos circuitos elétricos, o processo é semelhante: os de alta resistência conduzem corrente de menor intensidade que os de baixa resistência. A resistência de um componente elétrico depende do material que o constitui. Em geral, os fios são feitos de cobre, que oferece pouca resistência.

Cuidados básicos

Ao fazer qualquer reparo, lembre-se de desconectar o micro da tomada. Tome o cuidado também de anotar a posição inicial dos fios e dos componentes, a fim de evitar engano na hora da remontagem. Você pode marcar na própria placa utilizando caneta dermatográfica. No caso de um cabo multivias, por exemplo, marque a posição da faixa vermelha, indicadora do fio n.º 1. Para limpar os componentes e conectores tipo pente, use álcool isopropílico.



CUIDADO: seu computador perderá a garantia quando aberto por pessoas não credenciadas pelo fabricante.

Uma equação muito simples relaciona as três grandezas principais que entram no jogo algébrico dos circuitos elétricos e constituem a base para conhecer as funções dos componentes dos micros.

Georg Ohm, físico alemão que viveu no século XIX, foi o primeiro a deduzir uma equação matemática que relacionasse voltagem, corrente e resistência num circuito elétrico. Essa relação, chamada lei de Ohm, estabeleceu que a voltagem (tensão elétrica ou diferença de potencial) entre dois pontos de um circuito elétrico é igual ao produto da corrente pela resistência. Dessa forma, conhecendo duas das grandezas citadas, torna-se possível determinar a terceira: obtemos o valor da resistência dividindo a voltagem pela corrente; e calculamos o valor da corrente dividindo a voltagem pela resistência.

Usando os símbolos adotados por convenção, V para voltagem, I para corrente e R para resistência, essas relações podem ser representadas algebricamente de forma bastante simples:

$V = R \times I$
$R = V/I$
$I = V/R$

Uma extensão da lei de Ohm é a lei da potência:

$W = V \times I$

que relaciona W, a potência consumida num circuito ou dispositivo, com a voltagem entre seus terminais e a corrente que flui através dele. A potência é a energia despendida na unidade de tempo. Se a voltagem na fórmula anterior for expressa em volts e a corrente em ampères, a potência será calculada em watts.

O conhecimento dessas equações é indispensável para os que trabalham com circuitos elétricos. Mesmo em nível doméstico mostram-se de grande utilidade, pois permitem, por exemplo, determinar o tipo de fusível necessário a qualquer aparelho.

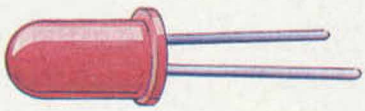
Como ilustração, considere um aquecedor elétrico que funcione em 220 V. Dividindo 3.000 (a potência, em watts, consumida pelo aparelho e indicada pelo fabricante) pela voltagem, a intensidade de corrente que atravessará o circuito será de 13,6 A. Desse modo, qualquer fusível de valor menor que 13,6 A se queimará imediatamente. Se o valor máximo suportável pela tomada for de 14 A, não se deve ligar a essa tomada nenhum aparelho que consuma mais do que 88 W — isto é, $(14 - 13{,}6) \times 220$ —, sob risco de comprometer todo o circuito.

Componentes

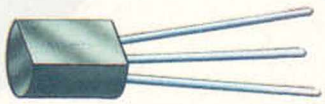
Os diagramas de circuitos eletrônicos parecem complicados. Mas, na verdade, constituem a forma mais simples de descrever os circuitos usando um conjunto padrão de símbolos e técnicas. Neste quadro, encontram-se todos os componentes de um intercomunicador e, ao centro, seu diagrama de circuito. Cada componente, nesse diagrama, é representado por um símbolo especial e identificado por um código: R1, TR2, C3. As linhas representam fios, ou trilhas numa placa de circuito impresso.

1. Chaves
Há dois tipos principais de chave: a permanente, cujo estado se mantém após ser pressionada, e a de pressão, em que o contato só permanece enquanto se mantiver a pressão.

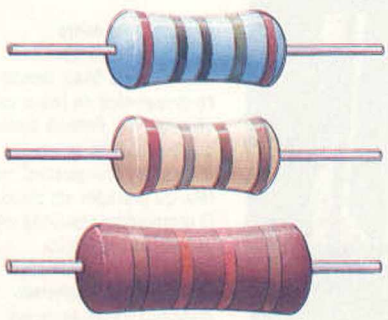
2. Diodos emissores de luz (LED)
Semicondutores de tipo mais simples, os diodos são o equivalente eletrônico das válvulas de retenção — só permitem a passagem da corrente num sentido. Alguns diodos, revestidos com resinas translúcidas em vez de metal, são usados como indicadores, pois emitem pequena quantidade de luz.

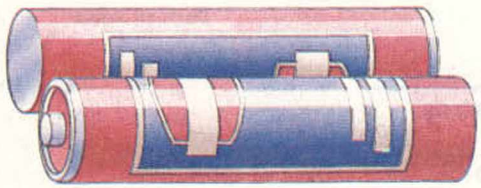


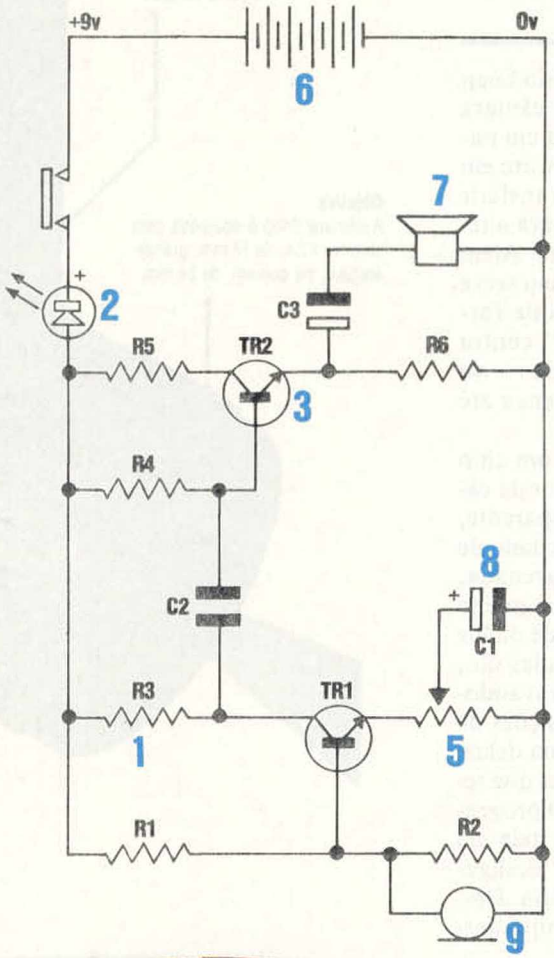
3. Transistores
Dependendo do tipo e da aplicação, o transistor pode agir como chave ou como amplificador. Sua invenção, em 1947, abriu o caminho para outros desenvolvimentos na microeletrônica.

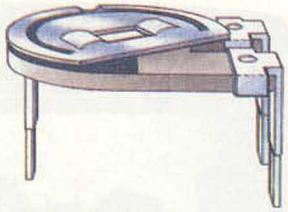


4. Resistores
Introduzidos nos circuitos, servem para controlar o fluxo de eletricidade. Há resistores de diversos tamanhos e seus valores são expressos em faixas coloridas.





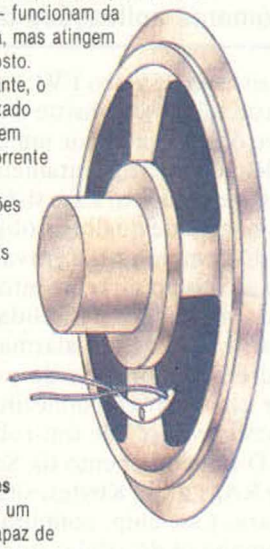




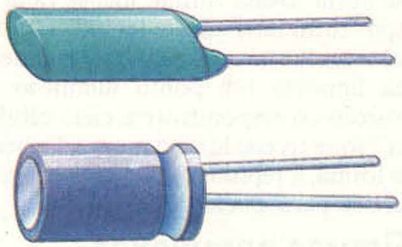
5. Resistores variáveis
Nem todos os resistores têm resistência constante. Alguns resistores variáveis, chamados de potenciômetros, usam uma tira de carbono como material condutor. O percurso da corrente pelo carbono determina a resistência do componente.

6. Pilhas
A maior parte dos circuitos pequenos, como o deste intercomunicador, pode funcionar com pilhas comuns, porque o consumo em watts do sistema é baixo e elas fornecem uma corrente contínua estável.

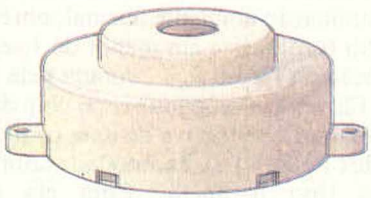
7. Alto-falantes
Alto-falantes são parentes próximos dos microfones — funcionam da mesma forma, mas atingem resultado oposto. Num alto-falante, o diagrama afixado ao ímã vibra em resposta à corrente aplicada. Essas vibrações produzem ondas sonoras no ar.



8. Capacitores
O capacitor é um dispositivo capaz de armazenar carga elétrica. Fica carregado quando se ligam seus terminais a uma fonte de energia e assim permanece mesmo depois de desligada a fonte. Para descarregá-lo é necessário conectar seus terminais entre si ou a outros condutores.



9. Microfones
Os microfones funcionam de maneira oposta à dos alto-falantes. Ondas sonoras fazem o diagrama vibrar e, em conseqüência, o ímã produz diferentes voltagens no circuito.



CÂMARA ESCURA

Do tamanho de um maço de cigarros e de fácil montagem, o sistema de vídeo EV1 reproduz na tela do computador as imagens captadas por uma câmara, proporcionando inúmeras aplicações de utilidade.

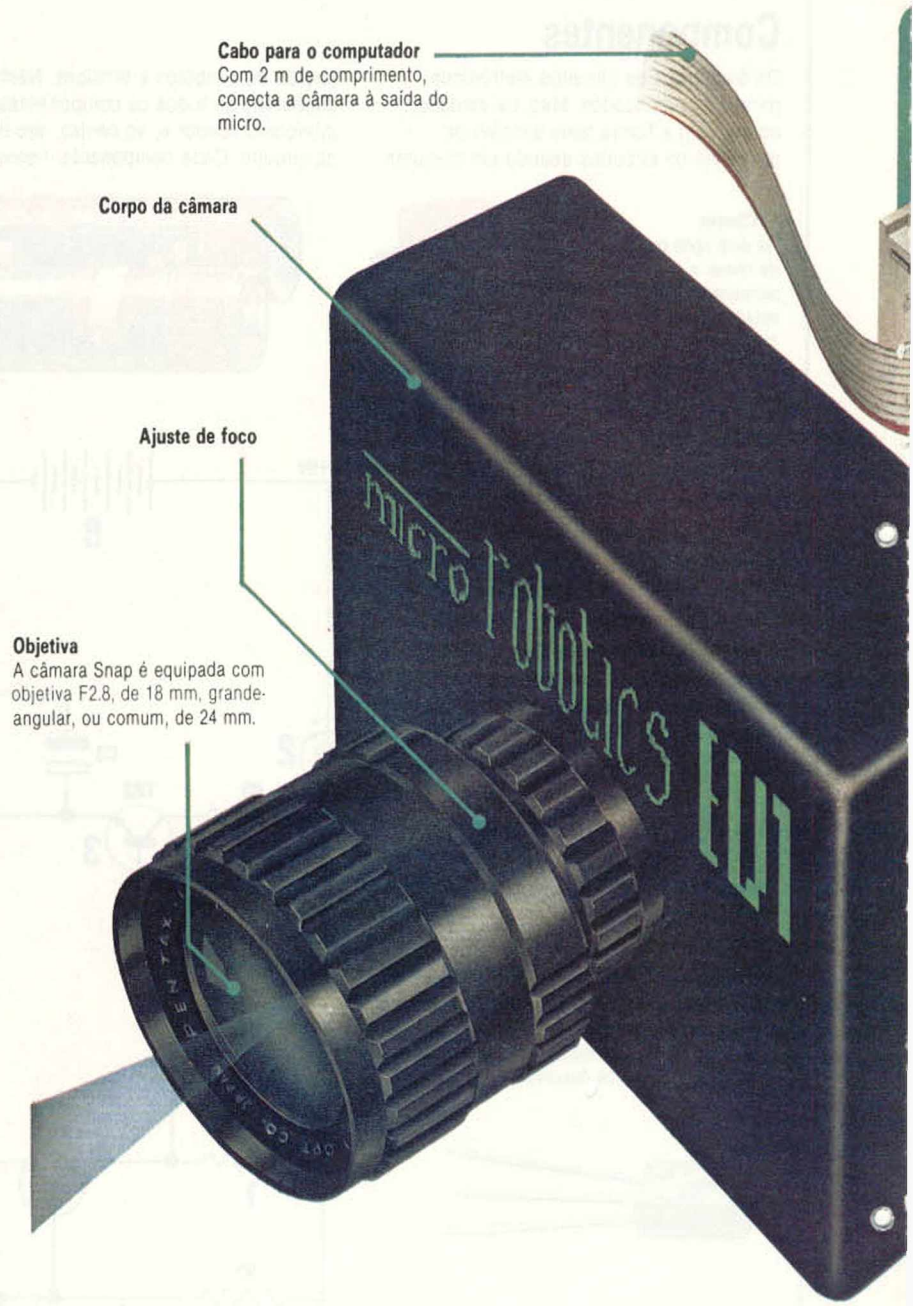
O sistema de vídeo EV1, também chamado Snap (instantâneo), consiste numa pequena câmara eletrônica ligada por um cabo à entrada em paralelo do micro, juntamente com o software em cassete ou disco. Esse sistema permite transferir a imagem de qualquer objeto ou cena para a tela do computador e gravá-la em disquete. Além do uso como divertimento, a câmara Snap serve para aplicações de utilidade prática: pode formar a base de um alarma "inteligente" contra ladrões — por meio do programa Secure — ou ser usada para reconhecimento de imagens e até como "visão" de um robô.

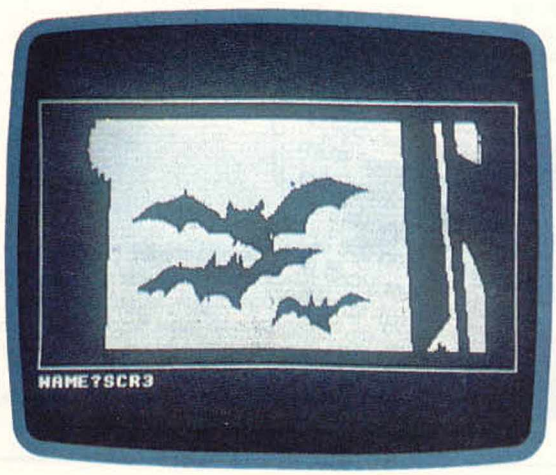
O funcionamento da Snap se deve a um chip de RAM de 32 Kbytes, situado no interior da câmara. Esse chip, com uma janela transparente, compõe-se de células que têm a propriedade de perder aos poucos a carga nelas armazenada, uma vez exposta à luz. A velocidade com que se descarregam é proporcional à intensidade da luz incidente. Na câmara Snap todas as células são, de início, completamente carregadas, gravando-se um valor específico em todas as posições de memória. Dessa forma, liga-se cada uma delas. Após curto intervalo de tempo, as células que receberem luz passam a descarregar-se. O programa imprime um ponto luminoso na tela na posição correspondente a cada célula "desligada" (que recebe luz) na matriz de memória. Dessa forma, a reprodução da imagem no chip transfere-se para a tela do micro.

Figuras complexas

O intervalo de tempo que o computador gasta ligando todas as células e depois lendo-as determina a "exposição" da imagem. A câmara Snap pode produzir várias imagens por segundo, quando a iluminação é intensa o suficiente; sob iluminação ambiente normal, obtém-se uma imagem fotográfica em menos de 1 segundo. A resolução da figura fica limitada pela matriz de 256 x 128 células de memória no chip da câmara.

A Snap usa a objetiva de uma pequena câmara reflex Pentax 110, acomodada sobre uma armação tipo baioneta. Com ela podem-se produzir figuras de dois tons a partir de uma só





Animação no micro
O EV1, software que acompanha a Snap, mostra continuamente na tela o que a câmara "vê". Permite também que as imagens sejam impressas, "congeladas" na tela, ou gravadas em disco. O computador seleciona um período de exposição conveniente, embora esse tempo possa ser alterado pressionando-se as teclas do cursor.

SCOPUS

HÁ 10 ANOS DESENVOLVENDO TECNOLOGIA BRASILEIRA

No Brasil, a era da informática começou recentemente. Mas não para a SCOPUS que, desde 1975, trabalha em projetos, produções, criações e fabricações das mais avançadas, destacando-se no mercado porque aposta no talento e capacidade de técnicos, engenheiros e especialistas brasileiros. Não mede esforços. Sempre à procura de soluções modernas e criativas, desenvolve o que há de mais arrojado em matéria de informática. Sem copiar o que é feito fora. Conheça agora algumas das atuais conquistas tecnológicas da SCOPUS.

Tecnologia e sofistic

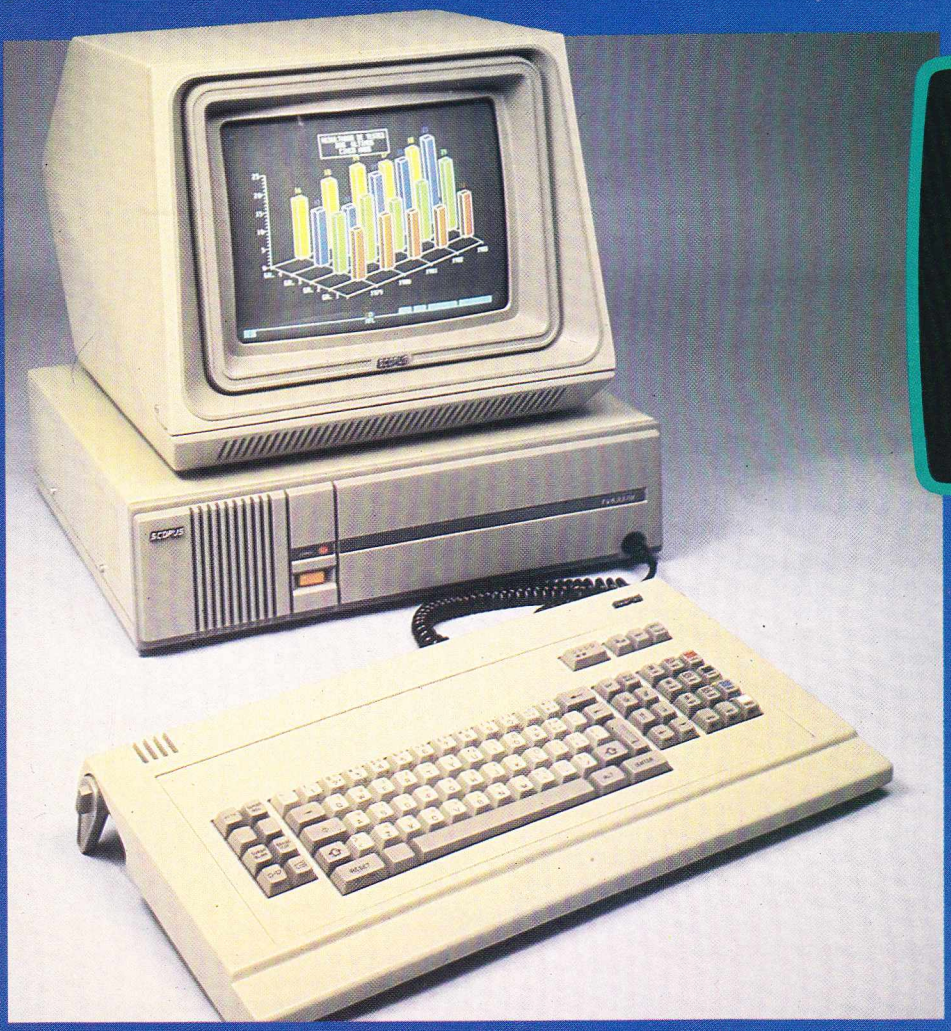
NEXUS 1684

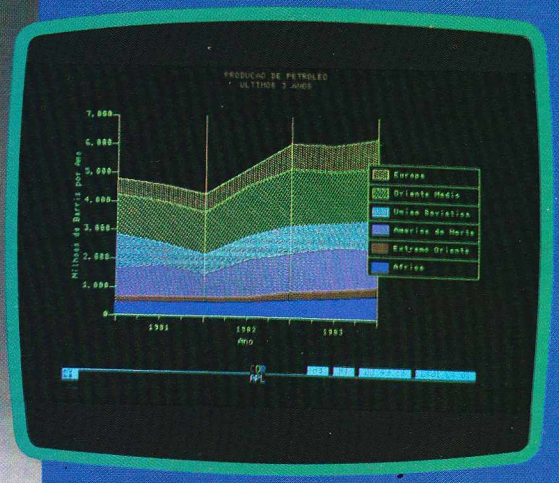
Como o mais recente integrante da família NEXUS 1600 de microcomputadores, o modelo 1684 traz uma importante novidade: a integração da unidade de disco magnético "Winchester", capaz de armazenar 10 milhões de caracteres (10 Mbytes). Dirigido a aplicações profissionais, o NEXUS pode ser utilizado para diversas operações, tais como análise financeira, análise de orçamento, balancete, entre outras tarefas gerenciais e administrativas. Ele se integra também a vários sistemas de grande porte, como computadores IBM, Burroughs ou Honeywell. Oferece características exclusivas, como o "clock" de CPU de 8 MHz e o tratamento de vídeo sem cintilação. Por trabalhar junto com a unidade Winchester, foi possível obter-se uma grande redução de custos além de um melhor aproveitamento do espaço físico.

TVA 3379

Com um hardware sofisticadíssimo e tecnologia (brasileira, é claro) das mais avançadas, o terminal de vídeo TVA 3379 tem como proposta principal facilitar a interpretação de dados fornecidos pelo computador através de gráficos a cores. É o único terminal desse tipo disponível para sistemas IBM existente no mercado. Usando o recurso das cores, o TVA 3379 facilita a leitura e inter-

ção em três versões.





pretação de qualquer informação, o que antes só poderia ser obtido através de tabelas numéricas. É a SCOPUS atenta a todo avanço e tendência do mercado mundial, facilitando a relação entre o homem e a máquina. Apenas usando o talento, o esforço e a criatividade de centenas de engenheiros brasileiros.

PC 2140

O mais novo e econômico "tradutor de linguagens", criado e desenvolvido com tecnologia totalmente nacional, o processador de comunicação PC 2140 permite a conexão simultânea de 32 equipamentos (terminais ou microcomputadores) com um computador de grande porte, através de uma única linha BSC-3. Isto permite o acesso ao banco de dados dos sistemas de grande porte, como IBM ou Burroughs, possibilitando ao usuário levantar importantes informações disponíveis no computador, entre elas, saldos de conta bancária, consultas ao pregão da Bolsa de Valores e verificação do estoque. O PC 2140 pode ser ligado a Unidades de Controle de Transmissão IBM 270X, 370X e ICA. A ligação dos equipamentos ao PC 2140 deve ser direta ou via modem, podendo-se também utilizar o modem de resposta automática.

Qualidade e assistência. A Scopus garante.

A avançada tecnologia do NEXUS 1600, PC 2140 e TVA 3379 já garante ao consumidor alto grau de qualidade e confiabilidade. Mas não é só isso que a SCOPUS oferece. Todos os serviços de manutenção são prestados por equipes de profissionais de alto nível técnico, em constante treinamento, que atendem rapidamente a qualquer chamado. Sempre investindo no aperfeiçoamento de sua estrutura de atendimento, a SCOPUS garante a mais elevada disponibilidade a seus equipamentos e a certeza de uma assistência técnica permanente.

DADOS TÉCNICOS

NEXUS 1600

UNIDADE DE SISTEMA

Placa de sistema: Microprocessador Intel 8088 (4,77 ou 8 MHz) • 64 Kbytes de memória EPROM • 256 Kbytes de memória RAM • 2 interfaces de comunicação serial: assíncrona e síncrona/assíncrona, padrão RS-232C, em velocidades até 19200 bauds • 5 conectores para placas de expansão.

Placa controladora de disco: controla duas unidades de disco flexível de 5 1/4", possui relógio-calendário de tempo real não volátil e duas interfaces para "joy-sticks".

Placa controladora de vídeo: modo alfanumérico: telas de 40x25 ou 80x25 caracteres, 16 cores • modo gráfico: telas de 320x200 pontos a 4 cores ou 640x200 pontos monocromáticas, interface para "light-pen".

Placa controladora de Winchester: controla a unidade de disco rígido tipo Winchester.

Unidade de disco flexível de 5 1/4": um acionador com cabeça dupla, densidade dupla, com capacidade de 320 Kbytes.

Unidade de disco rígido tipo Winchester: um acionador com capacidade de 10 Mbytes, slim.

MONITOR DE VÍDEO

Monocromático, 12", fósforo verde, interface padrão RGB, média resolução e configuração multinível, reproduzindo a informação da cor em tons de intensidades variáveis. Opcionalmente, pode-se configurar o Nexus 1600 com um monitor de vídeo cromático, 14", média resolução.

TECLADO

Tipo máquina de escrever, 86 teclas.

SOFTWARE BÁSICO

O Sistema Operacional SISNE, desenvolvido pela Scopus, emula o sistema MS-DOS 1.1 do microcomputador PC-IBM e contém todos os utilitários compatíveis com os utilitários do MS-DOS 1.1. Possui uma interface homem-máquina bastante poderosa e gerencia, com máxima eficiência, os recursos físicos do equipamento.

MÓDULOS DE EXPANSÃO

N1664: expansão de memória RAM (448 Kbytes) • **N1667**: co-processador CP-8087 • **N1669**: interface paralela para impressora.

MÓDULOS PARA COMUNICAÇÃO DE DADOS

Sistema Incox: interface coaxial para conexão a redes IBM 3270 • **NX-3270**: módulo para comunicação com sistemas IBM sob protocolo BSC-3 • **NX-830**: módulo para comunicação com sistemas Burroughs sob protocolo Poll-Select • **NX-VIP**: módulo para comunicação com sistemas Honeywell Bull sob protocolo VIP 7700 • **NX-Z**: módulo para comunicação assíncrona serial de dados.

TVA 3379

MÓDULOS

Unidade de vídeo: apresenta tela útil de 1920 ou 2560 posições, respectivamente para os modelos 2 ou 3. Possui duas linhas adicionais de informações no vídeo para mensagens do sistema e de erros, além de possibilitar a apresentação de cronômetro e indicador de linha e coluna. O monitor é cromático, de alta resolução. O cinescópio, de 14 polegadas, pode ser configurado para que se apague automaticamente, evitando a queima desnecessária do fósforo.

Unidade lógica: comporta as placas de controle do terminal, de geração de caracteres plano triplo, e as interfaces de comunicação com o vídeo, unidade controladora e teclado.

Teclado: comunica-se com a unidade lógica de forma serial e permite a conexão de um dispositivo auxiliar de entrada. Está disponível em quatro opções: APL Português, APL Inglês, "Typewriter" Português e "Typewriter" Inglês, todas com bloco destacado de teclas de cor e de símbolos programáveis.

PC 2140

CARACTERÍSTICAS FUNCIONAIS

Interface de comunicação com terminais ou micros: serial, assíncrona, código ASCII, full duplex, padrão elétrico RS-232C.

Placa interface de comunicação: cada placa permite que até 8 dispositivos sejam conectados ao PC 2140 com seleção de velocidades independentes.

Conexão de modem: a interface de comunicação com os terminais ou micros permite a ligação de modem, podendo ser constante ou por meio de linhas discadas, através de centrais telefônicas.

Interface de comunicação com o computador: serial, síncrona, sob o protocolo BSC-3, padrão elétrico RS-232C, em velocidades até 9600 bps. Emula as unidades de controle tipo IBM 3271 ou 3274 com 32 terminais IBM 3277 ou 3278, modelo 1.

Conversão de códigos: adotados pelos terminais (ASCII) em códigos adotados pelo computador (EBCDIC).

Execução de comandos: permitem ao computador ativar funções como, por exemplo, estabelecer conexões com os terminais, desfazer conexões, transmitir textos e informar o estado de ligações.

Verificação de paridade: em todas as mensagens com o computador, o PC 2140 gera e verifica os caracteres de controle de bloco (CRC-16), pedindo retransmissão automática em caso de erro.

MODELOS

O PC 2140 está disponível em quatro opções, que diferem apenas em relação ao número de placas de controle dos terminais.

PC 2140/8: controla até 8 terminais (1 placa).

PC 2140/16: controla até 16 terminais (2 placas)

PC 2140/24: controla até 24 terminais (3 placas)

PC 2140/32: controla até 32 terminais (4 placas)

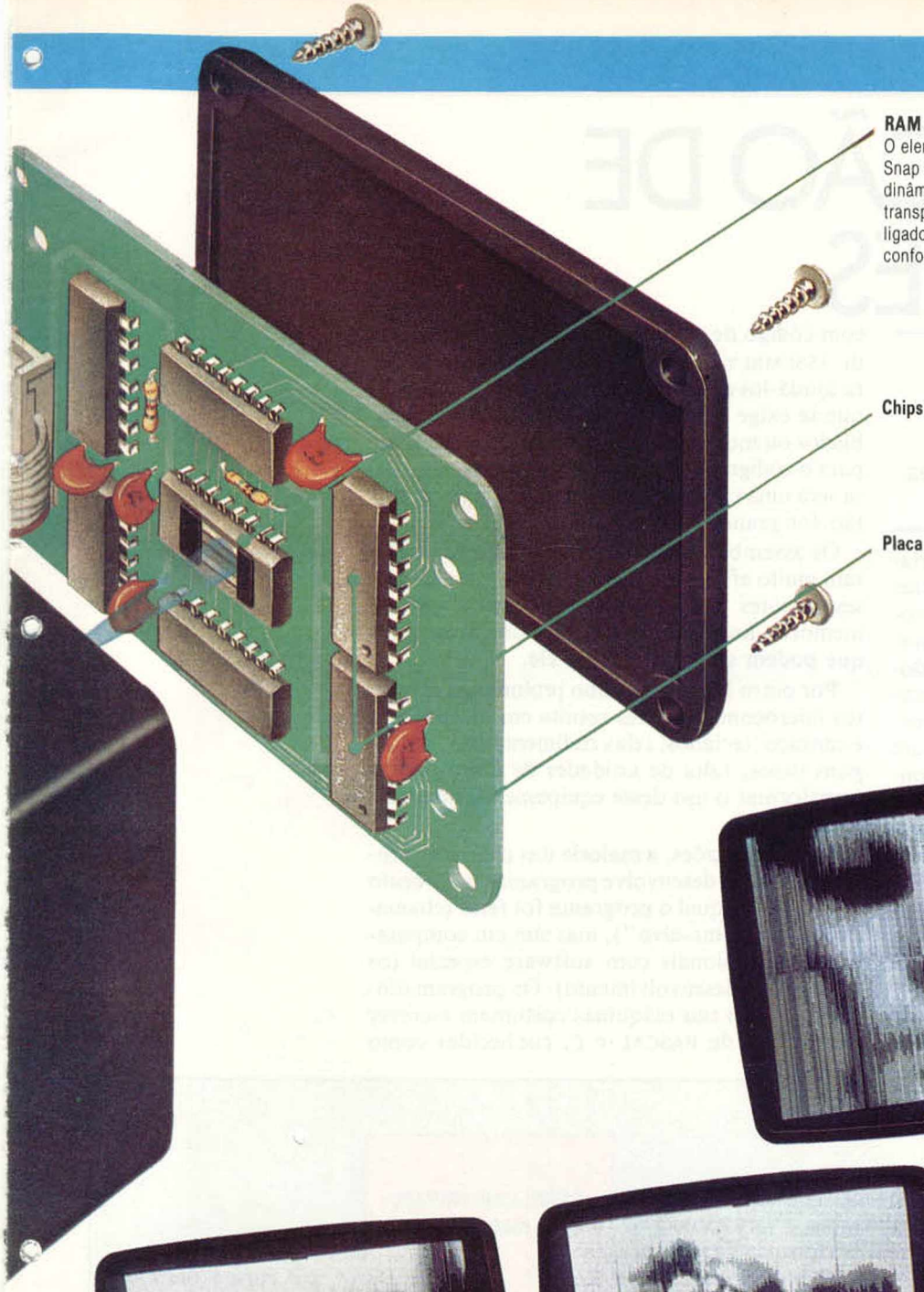
CENTROS DE ASSISTÊNCIA TÉCNICA

BELÉM: Tel.: (091) 228-0011 - ramal 242
BELO HORIZONTE: Tel.: (031) 224-8457
BRASÍLIA: Tels.: (061) 224-9856 e 224-9835
CAMPINAS: Tel.: (0192) 42-7333 - BIP: 657
CUIABÁ: Tels.: (065) 322-0542 321-5792 - BIP: 139
CURITIBA: Tel.: (041) 262-5211 - BIP: 150
FORTALEZA: Tel.: (085) 224-7999 - BIP: DA2
LONDRINA: Tel.: (0432) 22-0558
PORTO ALEGRE: Tel.: (0512) 32-1200 - BIP: 798
RECIFE: Tel.: (081) 221-3100 - BIP: 765
RIBEIRÃO PRETO: Tels.: (016) 636-3967 634-3331/3509 - BIP: 76B
RIO DE JANEIRO: Tels.: (021) 240-4925 e 240-6114
SALVADOR: Tel.: (071) 233-4117 - BIP: D12
SÃO JOSÉ DOS CAMPOS: Tels.: (0123) 22-8247 21-2004 - BIP: 536
SÃO PAULO: Tel.: (011) 255-7744

SCOPUS tecnologia s.a.

Filiada à ABICOMP

As especificações contidas neste folheto podem ser alteradas pelo fabricante sem aviso prévio.



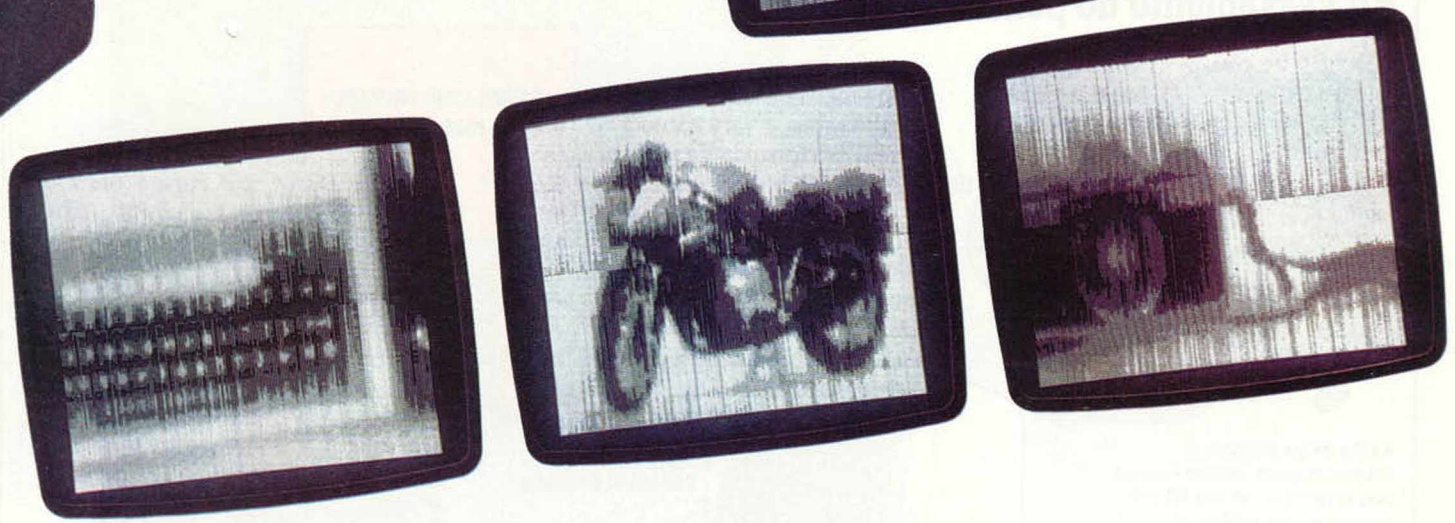
RAM óptica
O elemento fotossensível na Snap é um chip de RAM dinâmica, com janela de cristal transparente. Os bits são ligados ou desligados um a um conforme a incidência de luz.

Chips controladores

Placa de circuito

CÂMARA SNAP/EV1
Dimensões: 70 x 50 x 25 mm.
Objetiva: Lentes de 24 ou 18 mm.
Interface: Paralela.
Documentação: Manual de cinqüenta páginas, explicando o funcionamento da câmara e a utilização dos aplicativos.

Resolução vertical
As figuras geradas pela Snap são formadas por linhas verticais, com lacunas que indicam variações na luminosidade. A resolução das imagens é apenas aceitável.



imagem, ou de vários tons, tirando-se diversas fotos em exposições diferentes — caso em que se obtêm figuras mais realistas.

O primeiro método é usado por um dos programas fornecidos com o aplicativo, que inclui também uma rotina de dump de tela apropriada para impressoras Epson ou compatíveis.

O programa Movie armazena uma pequena seqüência de imagens em dois tons que são mostradas rapidamente, dando o efeito de animação. Talvez a opção mais atraente seja o programa chamado Arty, usado para construir figuras complexas a partir de diferentes imagens. Com um joystick, as imagens captadas pela câmara podem ser posicionadas em qualquer lugar da tela, tendo seu tamanho ampliado ou reduzido. As cores da imagem e do fundo são mudadas por meio das teclas de função.

Claro-escuro
As figuras aparecem geralmente em preto e branco. O programa Grey permite compor as imagens em tonalidades cinza, usando oito variações de brilho. O aumento do contraste por meio do sombreado oferece mais realismo à imagem.

A PRODUÇÃO DE SOFTWARES

Muitos amadores sonham desenvolver programas campeões de vendas. Mas a concorrência é forte demais: empresas fabricantes de softwares contam com enormes recursos técnicos para criar seus best sellers.

Um equipamento caro não implica necessariamente programas de sucesso; alguns amadores conseguiram muito dinheiro com softwares produzidos em casa, num micro de poucos recursos.

Ainda assim, os gênios da programação doméstica estão se tornando uma espécie ameaçada de extinção, sobretudo com o desenvolvimento das grandes empresas de software (software houses) na década de 80. Os poderosos computadores e sofisticados recursos de programação dessas empresas representam uma vantagem real sobre o proprietário de micro doméstico e possibilitam a seus programadores mostrar-se mais produtivos.

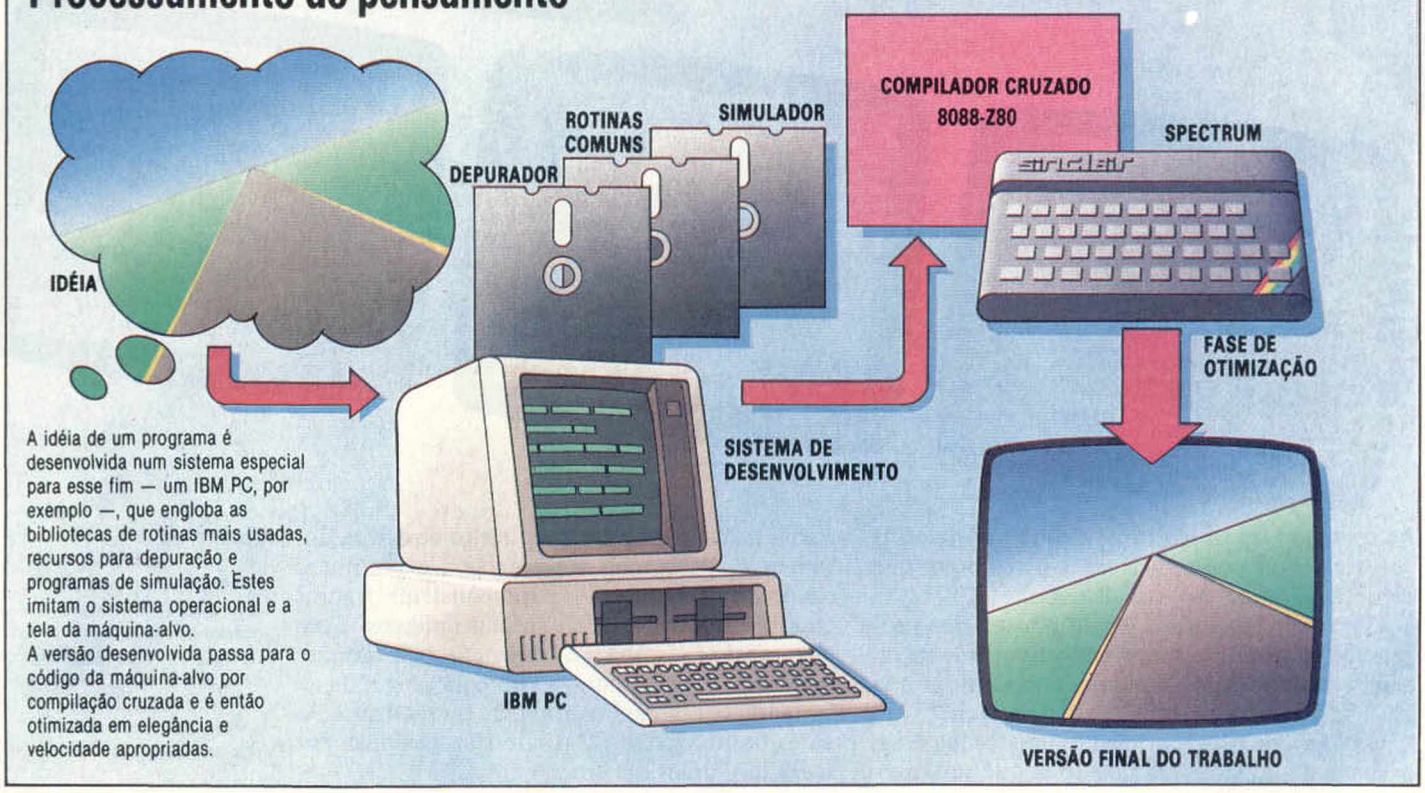
Uma característica importante do software para microcomputadores é a velocidade de execução. Isso significa que os programas precisam ser escritos (pelo menos parcialmente) em código de máquina (ASSEMBLY). Mas é difícil trabalhar com código de máquina — e os programadores de ASSEMBLY precisam de outros softwares para ajudá-los a escrever os programas. O mínimo que se exige é um programa assembler (assemblador ou montador) que traduza o código-fonte para o código-objeto que a máquina entende. Essa será uma tarefa difícil, se o programa em questão for grande.

Os assembladores para micros não se revelaram muito eficientes. Mesmo o mais simples desses pacotes usa quantidade considerável de memória, limitando o tamanho dos programas que podem ser escritos com ele.

Por outro lado, o trabalho prolongado em certos microcomputadores resulta em desconforto e cansaço: teclados, telas rudimentares e, em alguns casos, falta de unidades de disco podem transformar o uso desse equipamento numa tarefa árdua.

Por essas razões, a maioria das empresas profissionais não desenvolve programas no próprio micro para o qual o programa foi feito (chamado de "máquina-alvo"), mas sim em computadores profissionais com software especial (os sistemas de desenvolvimento). Os programadores que usam tais máquinas costumam escrever em versões de PASCAL e C, conhecidas como

Processamento do pensamento



A idéia de um programa é desenvolvida num sistema especial para esse fim — um IBM PC, por exemplo —, que engloba as bibliotecas de rotinas mais usadas, recursos para depuração e programas de simulação. Estes imitam o sistema operacional e a tela da máquina-alvo.
A versão desenvolvida passa para o código da máquina-alvo por compilação cruzada e é então otimizada em elegância e velocidade apropriadas.

"compiladores cruzados" ou "assembladores cruzados".

Conseguem, assim, desenvolver num micro que use um processador 8086, por exemplo, programas que serão executados em máquinas com processadores Z80. Numa fase posterior, especialistas em código de máquina revêem e melhoram os programas assim desenvolvidos.

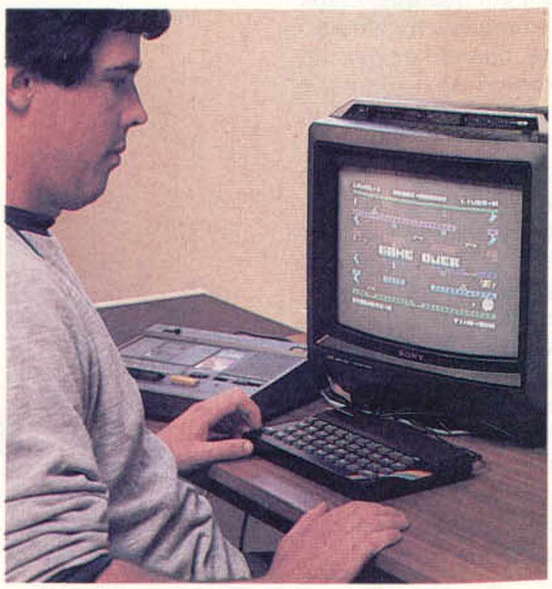
Softwares inteligentes

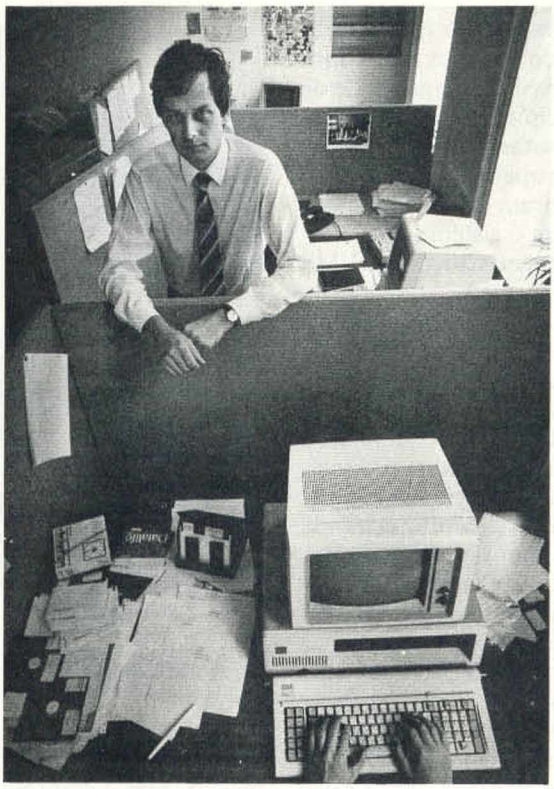
Rotinas de depuração de erros são acrescentadas sem a preocupação de o programa não caber na memória.

Algumas empresas européias podem exemplificar a técnica de desenvolvimento de programas. Uma delas é a Intelligent Software (IS), fundada em 1981, resultante de uma sociedade entre David Levy, especialista em xadrez, e a ANT Microwave, de Robert Madge. A IS especializou-se em jogos de estratégia, dos quais a maioria é escrita sob contrato para os micros mais populares. Desenvolveu também o software para máquinas que jogam xadrez.

Embora não haja telas de ação em jogos como o xadrez e o bridge, muita computação está envolvida em seu funcionamento. Assim como acontece com os jogos de fliperama, os de estratégia precisam da velocidade do software escrito.

Além de usar as próprias máquinas-alvo para desenvolver os programas, a IS vale-se de micros IBM PC e Apple com interfaces especialmente desenvolvidas para transferir programas de uma máquina para outra. A empresa está constantemente envolvida em projetos de conversão — adaptando, por exemplo, um jogo de xadrez de um tipo de computador para outro —, e assim é possível programar de forma modular. Por exemplo, um nível de segmentação útil quando se tem de converter programas de jogos de um processador para outro é a divisão do programa em rotinas de jogo e rotinas de entrada/saída. Estas, na máquina que está recebendo o programa, têm endereços de memória diferentes e talvez também sejam diferentes em termos de endereçamento de periféricos. Rotinas de jogos, por serem independentes do hardware (exceto o processador), têm conversão direta.





Módulos universais
O desenvolvimento de jogos de estratégia (o xadrez, por exemplo) é feito na máquina-alvo ou em máquinas mais poderosas, como o IBM PC. Programas divididos em módulos universais podem ser facilmente adaptados de um computador para outro

A Intelligent Software evita cercear a criatividade dos programadores; assim, há poucas "regras da casa" direcionando a programação. Um ponto importante no qual ela insiste, contudo, é que o programa-fonte deve incluir numerosos comentários, de modo que seja sempre claro o que faz cada rotina.

Quando os programadores trabalham em suas casas para uma empresa de software, cada um desenvolvendo projeto próprio, há muito pouco compartilhamento de recursos. Em decorrência, a individualidade preserva-se às custas da multiplicação de esforços, pois rotinas semelhantes acabam sendo escritas e reescritas por vários programadores.

Hardware pesado

Outra empresa de software, a Psion, utiliza computadores ainda maiores que o IBM PC. Entre as empresas de software inglesas que desenvolvem jogos para microcomputadores, a Psion é a única a concentrar suas pesquisas de desenvolvimento em minicomputadores.

Ela começou desenvolvendo softwares para o ZX81, um pequeno micro da Sinclair — e usava o próprio ZX81 para fazê-los. Para criar a fita Horizons, que acompanha o Spectrum, a Psion comprou um TRS-80 com discos, que usa o mesmo processador Z80, e criou uma interface especial entre as duas máquinas.

Mas, em agosto de 1982, a empresa decidiu que não poderia continuar criando sistemas de desenvolvimento completamente diferentes cada

Uso da máquina-alvo
A partir da definição da estratégia e do roteiro do jogo na máquina-alvo, o programa é desenvolvido em ASSEMBLY.

vez que um novo microcomputador era lançado no mercado. Então, aplicou recursos na compra de hardware pesado com bastante capacidade de processamento. Em princípio, esses hardwares deveriam ser flexíveis o bastante para enfrentar qualquer computador que aparecesse no futuro. As máquinas escolhidas foram dois minicomputadores Vax 750, que usam o sistema operacional DEC (Digital Equipment Corp) em VMS (Virtual Machine System).

Os Vax 750 trouxeram duas vantagens para a Psion: a qualidade de software proporcionada pelo DEC, com a oportunidade que ele dá de criar recursos auxiliares de software especialmente projetados, e a "força bruta" da combinação do sistema operacional com o hardware. Há muito espaço para uma coleção de recursos de software, como compiladores, bibliotecas de sub-rotinas comuns e programas de depuração, todos compartilhados até entre vinte programadores, que podem estar acessando uma única máquina ao mesmo tempo. As duas máquinas permitem que o software seja convertido de uma para outra, quando necessário, com facilidade.

As bibliotecas de sub-rotinas comuns já faziam parte da filosofia da Psion no tempo do TRS-80, mas ficar trocando dados entre dois discos é trabalhoso. Com as máquinas Vax, equipes de programadores podem até trabalhar juntas, compartilhando bibliotecas de projetos, das quais se conseguem módulos quase instantaneamente. Essas bibliotecas são até compartilhadas por equipes ocupadas em projetos diferentes.

Essa é a grande vantagem de um sistema com tempo compartilhado — e, como vantagem adicional, ele também cuida do trabalho administrativo sem ter de interromper os programadores.

Mesmo se você tem condições para isso, não pense que, comprando um Vax, ficaria imediatamente em igualdade de condições com uma firma como a Psion. Uma parte muito pequena desse ambiente de trabalho bem desenvolvido foi entregue numa bandeja à Psion pelo DEC. A empresa despendeu muito esforço para conseguir que tarefas simples fossem realizadas com eficiência e confiança e para chegar ao grande número de recursos de software e utilitários elaborados à mão (escritos em C) que acumulou.

A Psion usa C, linguagem de grande portabilidade que pode produzir um código-objeto compacto e rápido para chips de 16 bits, como o 8086. Mas isso não se aplica a compiladores C de 8 bits. Assim, ao escrever para máquinas-alvo como o Spectrum, a empresa precisou desenvolver suas próprias técnicas especiais.

Sabe-se que a Psion usou C para escrever seu próprio compilador. Ele se parece um pouco com o C, é transferível para diferentes processadores e cria programas muito eficientes.

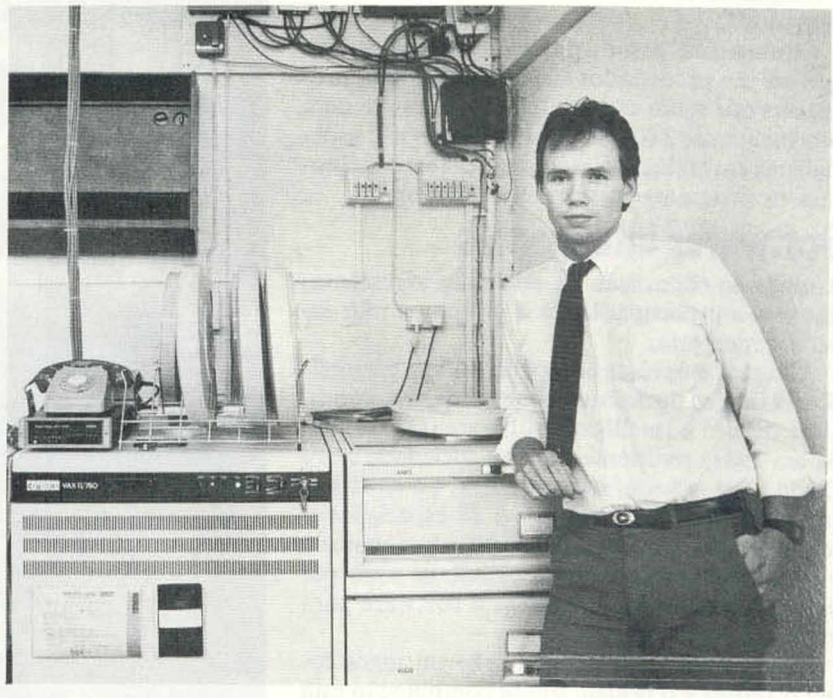
Código-fonte

Há uma regra segundo a qual a manutenção dos sistemas e o trabalho de escrever recursos auxiliares de programação (como o compilador da Psion) ocupam 30% do tempo total de programação. Mas a empresa acha que esses homens-hora a mais valeram a pena. Desenvolver seu próprio código-fonte significa ter domínio absoluto sobre ele: você pode desmontá-lo e aperfeiçoá-lo ou adaptá-lo, o que é impossível com o software adquirido no comércio.

O software especial comprado pela Psion inclui programas que são simulações exatas dos microprocessadores mais comuns, como o Z80 e o 6502. Assim, pode-se fazer com que os computadores Vax se comportem como se fossem um Commodore 64 ou um Spectrum. Apesar da grande capacidade dos computadores Vax, os simuladores fazem-nos funcionar numa fração da velocidade da máquina-alvo.

A vantagem é que eles permitem que o programador examine o conteúdo de cada registro dentro do microprocessador em qualquer etapa do programa, facilitando sobremaneira a detecção de erros. Normalmente, quando ocorre um erro num programa em código de máquina, é quase impossível dizer o que saiu errado. A Psion poupa, assim, muitas horas de depuração.

A maior parte do esforço de desenvolvimento da Psion foi canalizada para a produção do conjunto de quatro programas comerciais que vêm com o Sinclair QL. A família de chips Motorola 68000, um dos quais é responsável pelo funcionamento do QL, foi projetada com base nas linguagens de alto nível. Os programas em C são compilados com tanta eficiência, com esses chips, que se torna desnecessário escrever em ASSEMBLER. Se todos os computadores de uso doméstico seguissem o exemplo do QL, o C substituiria por completo o ASSEMBLY e a Psion e as empresas menores de software poderiam abandonar o trabalho extenuante de fazer conversões à mão.



Exceção brasileira
Softwares para micros podem resultar do trabalho de vários programadores que compartilham um equipamento maior, como o Vax 750. No Brasil, contudo, é raro o uso de "compiladores cruzados". Alguns fabricantes de micros que desenvolvem seu próprio software constituem uma exceção.

EM CONJUNTO

Além das estruturas de dados já vistas, como matrizes e arquivos, o PASCAL inclui os conjuntos, um tipo estruturado de dados que será examinado a seguir, juntamente com os níveis de prioridade dos operadores.

Um conjunto nada mais é do que uma coletânea de objetos processados em bloco, como uma entidade singular, em vez de cada elemento ser acessado individualmente. Um exemplo seria o conjunto das pessoas que programam em PASCAL ou as letras do alfabeto que correspondem às vogais. Nem sempre os elementos de um conjunto estão ordenados e, nesse caso, a única questão relevante é saber se determinado objeto pertence ou não ao conjunto.

Para aumentar sua eficiência, o PASCAL impõe algumas restrições aos conjuntos. Seus membros só podem ser tipos escalares simples — e não matrizes ou outros dados estruturados — e há um limite máximo (definido pela implementação) para esse tipo "base". A sintaxe é simples:

```
TYPE
  NUMEROS = SET OF 0 .. 127;
  ALFAS = SET OF 'A' .. 'Z';
  CORES = SET OF (VERMELHO, VERDE,
    AZUL);
```

A faixa dos valores que o tipo base ordinal do conjunto pode assumir é definida por meio da mesma sintaxe usada para os tipos de subfaixa. As variáveis de conjunto são declaradas na parte da declaração VAR, do mesmo modo que as outras variáveis do PASCAL; e podem, nas instruções, vir por extenso entre colchetes (isto é, como "literais"):

```
VAR
  CODIGOS : NUMEROS
  LETRAS : ALFAS
BEGIN
  CODIGOS := [0 .. 2, 4, 8, 16, 32, 64];
  LETRAS := ['A' .. 'Z']; {ETC.}
```

Esse segmento inicializaria os códigos do conjunto, fazendo-os conter somente os números de 0 a 2 (inclusive), de 4 a 8, e assim sucessivamente. Como não há uma ordem inerente ao próprio conjunto (ao contrário de seu tipo base), poderíamos da mesma forma representar esse conjunto como [64, 32, 16, 8, 4, 0 .. 2], mas observe que a subfaixa 2 .. 0 (ilegítima na presente definição de subfaixa) iria meramente indicar uma faixa vazia. Qualquer conjunto pode ser inicializado como um conjunto vazio de seu tipo por meio da instrução QUALQUERSET := [], que origina a única exceção à regra geral do PASCAL de que se conhece por inspeção o tipo de qualquer literal. Se pelo menos um membro do conjunto não aparecer por extenso, não poderemos — nem o compilador — determinar seu tipo. Felizmente, o conjunto vazio pode ocorrer somente em atribuições (como no exemplo) ou em expressões nas quais os outros identificadores já foram declarados. Isso significa apenas que um conjunto vazio sempre é um subconjunto de qualquer conjunto.

Operadores de conjunto

Um dos operadores mais úteis de que o PASCAL dispõe para estruturas de conjuntos é, como DIV e MOD, uma palavra reservada: IN. Trata-se de um operador que relaciona dois operandos e verifica a presença de um elemento num conjunto específico. O lado da esquerda deve ser uma expressão que resulta em um dos elementos possíveis do conjunto — em outros termos, um valor do tipo base do conjunto — e o lado da direita pode ser uma variável de conjunto ou uma literal de conjunto. Toda a expressão dará um resultado booleano: verdadeiro, se o valor for membro do conjunto, e falso, em caso contrário.

Assim: N IN CODIGOS e X IN LETRAS são expressões booleanas legítimas. Para verificar se um caractere é um dígito, o teste

```
IF (C >= '0') AND (C <= '9') THEN...
```

é muito simplificado se testamos o valor de C, examinando se ele é membro do conjunto de caracteres em que estamos interessados:

```
IF C IN ('0' .. '9') THEN...
```

Ou, talvez, se estivermos desenvolvendo um programa para jogar cartas:

```
TYPE
  ROL = (DOIS, TRES, QUATRO, CINCO, SEIS,
    SETE, OITO, NOVE, DEZ, VALETE, DAMA,
      REI, AS);
  CARTAS = SET OF ROL
VAR
  CARTA : ROL
  FIGURAS : CARTAS
BEGIN
  FIGURAS : VALETE..AS);
  IF CARTA IN FIGURAS THEN {ETC.}
```

Compare esse exemplo com:

```
IF (CARTA = VALETE) OR (CARTA = DAMA)
  OR (CARTA = REI) OR...
```

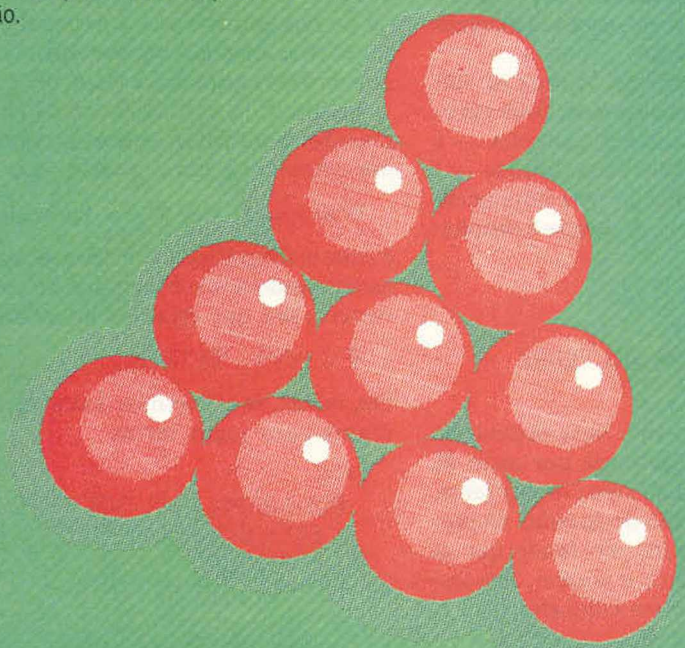
No PASCAL também se definem operações envolvendo conjuntos inteiros: interseção, união e diferença de conjuntos. Se B for o conjunto de todos os programadores em BASIC e P representar os programadores em PASCAL, a interseção dos dois conjuntos será o conjunto de programadores que usam tanto BASIC como PASCAL. A união de P e B é o conjunto de pessoas que programam ou em PASCAL ou em BASIC.

Tacada perfeita

O jogo de sinuca pode ser disputado com quinze bolas vermelhas (cada uma valendo 1 ponto), uma bola "jogadeira" branca e seis coloridas, que equivalem aos seguintes pontos: amarela, 2; verde, 3; marrom, 4; azul, 5; rosa, 6; e preta, 7.

Após cada bola vermelha ser encaçapada, o jogador pode testar sua habilidade com uma das outras cores. Estas, se encaçapadas, são a seguir recolocadas na mesa. Depois da última combinação de vermelho e outra cor, todas as cores devem ser encaçapadas em ordem crescente de valor.

Na sinuca, uma série de tacadas só termina: **a)** quando o jogador deixa de encaçapar uma bola; **b)** quando comete alguma falta; e **c)** quando não resta nenhuma bola (encerrando-se o jogo). Escreva um programa para calcular a maior série possível de tacadas, mas — e aqui está o truque — de tal modo que 15 seja o único número existente no programa.
A solução será apresentada na próxima matéria desta seção.



Prioridade de operadores

Esta tabela mostra a ordem de prioridade dos operadores do PASCAL. Se necessário, o uso de parênteses obriga à avaliação separada da expressão por eles delimitada, cancelando a ordem normal de prioridade.

Prioridade	Operadores	Tipo
Máxima	NOT + −	{monádica}
	AND * /DIV MOD	{multiplicação}
	OR + −	{soma}
Mínima	< <= <> >= > IN	{relacional}

A diferença de conjuntos é, como o nome sugere, o resultado de se retirar ou subtrair de um conjunto todos os membros de outro conjunto. Assim, na notação do PASCAL, P−B representa todos os que programam em PASCAL mas não em BASIC. Analogamente, a notação usada para a união será P+B, e para a interseção, P * B. Esses operadores são indicados pelos mesmos símbolos empregados para os operadores aritméticos, pois, nas operações com conjuntos, os operadores representam o teste dos bits envolvidos.

Imagine um conjunto de oito elementos. A presença de determinado elemento num conjunto poderia ser indicada colocando-se o bit apropriado em um grupo de 8 bits (1 byte); enquanto sua ausência no conjunto seria assinalada no byte por meio de um zero. Esses testes de inclusão exigem apenas uma verificação de bits: o conjunto união resulta de uma operação OR; e a interseção, de um simples AND. Essas operações são efetuadas em nível de código de máquina, de modo que o compilador do PASCAL possa fornecer estruturas de dados de conjuntos e suas operações correspondentes. O dispêndio geral de memória é mínimo e, quando o tamanho dos conjuntos pode ser mantido no limite do tamanho das palavras do computador (em máquinas de 32 bits ou mais, por exemplo), as operações com conjuntos são as desempenhadas com mais eficiência pelo PASCAL.

Ordem de prioridade

Podemos agora resumir todos os operadores usados em PASCAL. Aqueles que não foram citados são todos conhecidos de outras linguagens, mas no PASCAL tudo é simplificado, pois existem apenas quatro níveis de prioridade de operadores. Evidentemente, os operadores "monádicos" têm prioridade sobre os demais. São os símbolos + e −, que indicam o sinal dos números, e o operador de negação booleana NOT. O segundo nível de prioridade inclui todos os operadores de "multiplicação" (inclusive os símbolos de divisão) seguidos pelos de adição e subtração e, por fim, no nível mais baixo de prioridade, todos os operadores relacionais, inclusive IN.

Observe que os dois outros operadores booleanos (AND e OR) são operadores de multiplicação e de adição, respectivamente. Isso reflete a álgebra booleana utilizada e implica a colocação de muitos testes relacionais entre sinais delimitadores para cancelar essa prioridade. Caso contrário, IF N>0 AND N<10 THEN..., por exemplo, resultará em erro quando o programa for compilado, pois a expressão 0 AND N (avaliada em primeiro lugar) tenta combinar dois operandos integer com um operador booleano.

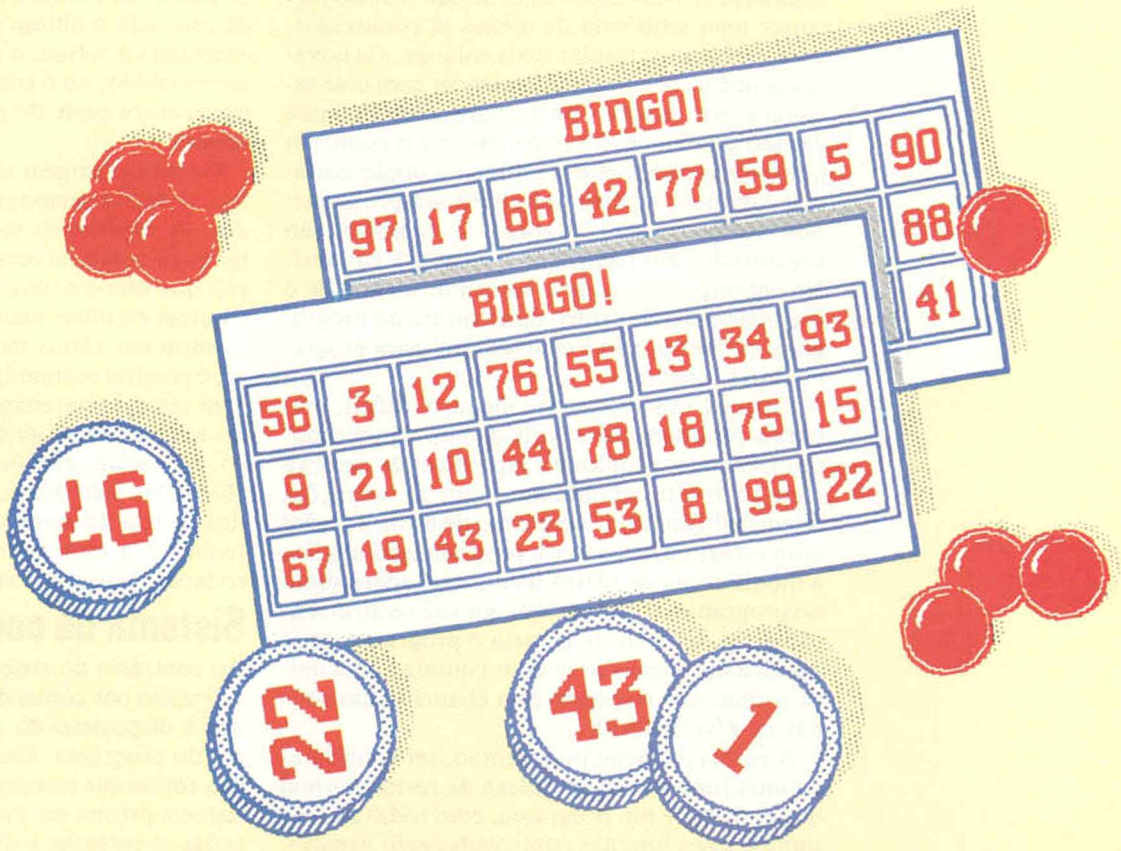
Quaisquer operadores com o mesmo nível de prioridade são avaliados, como de costume, da esquerda para a direita. O símbolo de atribuição (:=) possui uma prioridade mais baixa que qualquer outro dos operadores acima, pois a expressão do lado direito deve ser completamente avaliada antes de se fazer a atribuição.

Bingo!

Um conjunto é o melhor modo de se representar uma cartela de bingo. Embora os elementos (integers de 1 a 90) sejam ordenados, a única questão relevante é saber se o número chamado está ou não na cartela ou, em PASCAL, se "NUMERO IN CARTELA" é verdadeiro ou falso.

O programa simula um jogo de bingo lendo os números da cartela por meio do teclado e, depois, à medida que são chamados, verificando se correspondem ao conjunto da cartela. A expressão CARTELA – CHAMADOS será um conjunto vazio quando todos os números de CARTELA também estiverem no conjunto CHAMADOS. Note que não se pode incluir um membro diretamente no conjunto; só pela criação de um conjunto com um único elemento (colocando o número entre colchetes) podemos efetuar a união de dois conjuntos. Assim como está, o programa admitirá entradas duplicadas de cartelas e números fora da faixa de 1 a 90, causando um erro de execução. Como exercício, imagine um método para acrescentar estruturas que evitem essas duas irregularidades.

```
PROGRAM     BINGO     (INPUT, OUTPUT);

CONST
          COLUNAS = 40;  {PARA CONTROLAR VIDEO}
          METADE = 25;  {IDEM}

TYPE
          BINGO = SET OF 1 .. 90;

VAR
          CONTADOR          : 1 .. 15;
          POSSIVEIS,
          CHAMADOS,
          VAZIA,
          CARTELA           : BINGO;
          NUMERO            : INTEGER;
          FEITO             : BOOLEAN;


BEGIN

  VAZIA := [];
  POSSIVEIS := [1 .. 90];
  WRITELN ('*** BINGO ***'); METADE);
  WRITELN;
  WRITELN ('ENTRE COM OS 15 NUMEROS DA
    CARTELA');
  WRITELN ('(CADA UM SEGUIDO DE RETURN)');
  CARTELA := VAZIA;
  FOR CONTADOR := 1 TO 15 DO
    BEGIN
      WRITE (CONTADOR : 10, ' ?');
      READLN (NUMERO);
      CARTELA := CARTELA + [NUMERO]
    END;

  WRITELN;
  WRITELN ('OLHO VIVO!', METADE);
  WRITELN;
  WRITELN ('AGORA CHAME OS NUMEROS');
  CHAMADOS := VAZIA;

  REPEAT
    WRITE ('?', METADE);
    READLN (NUMERO);
    CHAMADOS := CHAMADOS + [NUMERO];
    FEITO := CARTELA - CHAMADOS = VAZIA

  UNTIL FEITO;

  WRITELN;
  WRITELN ('PARABENS!', METADE);
  WRITELN ('SEUS NUMEROS ERAM:');
  WRITELN;

  FOR NUMERO := 1 TO 90 DO
    IF NUMERO IN CARTELA THEN
      WRITE (NUMERO : COLUNAS DIV 8)

END.
```



FAÇA SUA ESCOLHA

Qualquer programa complexo exige que o usuário escolha, em determinados pontos, entre vários caminhos alternativos. Os sistemas orientados por menus ou por comandos são recursos que facilitam essas decisões.

Um menu pode ser tanto uma simples lista de itens numerados como uma tela cheia de complexos símbolos visuais, mas o princípio que rege seu uso é sempre o mesmo. Emprega-se o menu quando a lógica do programa chega a um ponto em que são possíveis vários caminhos alternativos e o usuário escolhe, a partir de uma lista de opções mostrada na tela, qual a rota a ser tomada. Os programas baseados em menus possuem, em geral, uma estrutura em forma de árvore: o usuário entra na árvore pela "raiz" e é guiado, pelas opções do menu, até uma das "folhas", onde a informação ou função específica se encontra.

A vantagem disso é não exigir do usuário nenhum conhecimento do programa, porque a rota é bem "sinalizada" em toda a sua extensão. Contudo, os mais experientes acham tedioso percorrer uma seqüência de menus já conhecidos, para executar as tarefas mais comuns. Os novatos também encontram dificuldades com uma estrutura em árvore, pois, para corrigir uma decisão errada, devem retroceder até o ponto em que o erro foi cometido, retomar a opção correta e continuar a partir dali. Nem sempre os menus formam uma árvore — também são organizados em rede, numa estrutura em anel. No entanto, isso tende a aumentar o risco de o usuário perder-se dentro da estrutura do programa e, portanto, não é aconselhável para programadores iniciantes.

Projetar um sistema de menus é difícil, embora a programação envolvida não seja complicada. O maior problema é especificar de maneira clara todo o programa, antes que as instruções sejam codificadas. O acréscimo de novas funções num estágio avançado da programação implica a modificação de vários menus que apareceram no programa, o que pode exigir sua reestruturação geral. Quando se projeta o programa, toda a lógica dos menus deve estar contida numa única rotina, que responde pela chamada das rotinas de nível inferior.

A rotina de menu pode, então, ser considerada uma forma mais complexa da rotina normal de controle de um programa, com todas as suas ramificações internas controladas pelo usuário. Isso assegura um bom nível de organização para o projeto e serve para separar a lógica de controle e as partes funcionais do programa, permitindo que cada uma seja desenvolvida e depurada em separado.

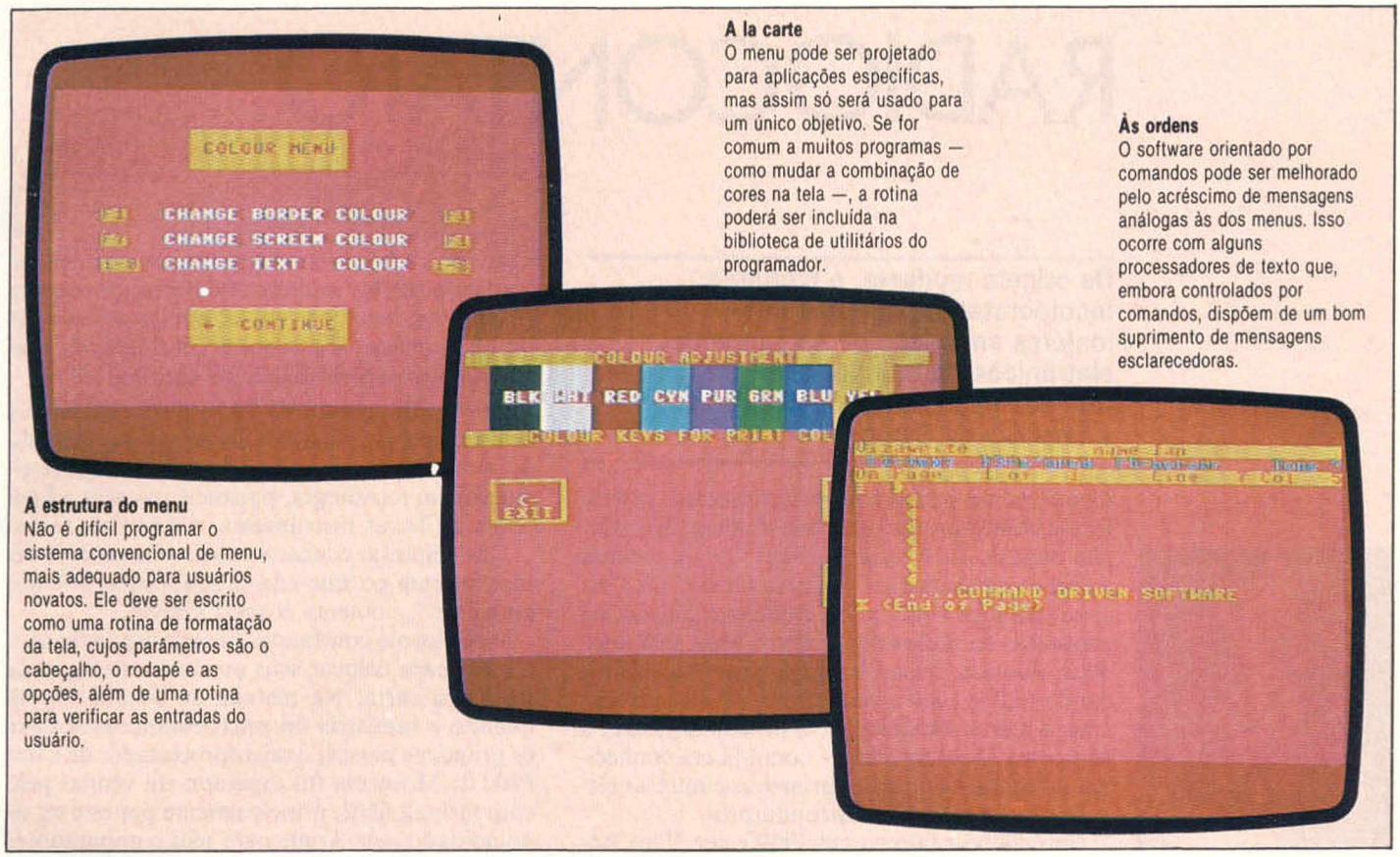
O fluxo de controle vai agora seguir um caminho predeterminado. Para cada menu ao longo da rota, a rotina da lógica do menu transfere um conjunto de mensagens ao usuário para outra rotina que as encaixa na "moldura" do menu. Em geral, este contém uma mensagem de cabeçalho, que mostra um título e quaisquer outras informações necessárias sobre o menu; um "rodapé", que explica como fazer a escolha (com espaço suficiente para a resposta do usuário); e as opções do menu propriamente ditas. Uma tela de menu eficiente tem no máximo oito opções dispostas em coluna, com os códigos de resposta (número, letras, código mnemônico etc.) à esquerda de cada item.

A rotina de menu chama uma rotina de entrada, passando-lhe as condições para entradas válidas, e aceita a resposta do usuário. Em seguida, a rotina de menu interpreta essa resposta (que em geral é dada pressionando-se uma única tecla) e passa o controle para o próximo nível de menu, ou chama a rotina aplicativa apropriada caso seja o último menu da cadeia. Uma vez executada a rotina, o menu que a chamou pode ser reexibido, ou o controle pode passar para alguma outra parte do programa (o menu inicial, talvez).

Os menus exigem muito texto para o cabeçalho, rodapé e mensagens, e muitos serão repetidos. A explicação sobre o funcionamento do menu (o modo de acesso às telas de ajuda, a opção que oferece uma saída para o menu inicial e outras escolhas mais freqüentes) pode ser necessária em vários menus diferentes. Nesse caso, é possível economizar espaço e tornar a lógica mais clara, armazenando todas as mensagens numa matriz alfanumérica (ou num arquivo em disco de acesso aleatório), da qual possam ser chamadas pelo número de índice. Projete a rotina de tela do menu de forma que ela aceite referências a essa matriz e mostre o cabeçalho, rodapé, mensagens etc. apropriados.

Sistema de comandos

Ao contrário do sistema de menus, um sistema orientado por comandos coloca todos os comandos à disposição do usuário em qualquer estágio do programa. Cada comando remete a uma sub-rotina que executa a função requisitada. Esse sistema precisa ser projetado de forma a checar todas as entradas e diferençar os dados dos co-



A la carte
O menu pode ser projetado para aplicações específicas, mas assim só será usado para um único objetivo. Se for comum a muitos programas — como mudar a combinação de cores na tela —, a rotina poderá ser incluída na biblioteca de utilitários do programador.

Às ordens
O software orientado por comandos pode ser melhorado pelo acréscimo de mensagens análogas às dos menus. Isso ocorre com alguns processadores de texto que, embora controlados por comandos, dispõem de um bom suprimento de mensagens esclarecedoras.

A estrutura do menu
Não é difícil programar o sistema convencional de menu, mais adequado para usuários novatos. Ele deve ser escrito como uma rotina de formatação da tela, cujos parâmetros são o cabeçalho, o rodapé e as opções, além de uma rotina para verificar as entradas do usuário.

mandos do programa. Assinala-se essa diferença pelo uso de determinada tecla antes de entrar com um comando. A tecla [Control] é freqüentemente usada para isso, assim como os caracteres ponto (.) e barra (/); uma vez que não são usados no início de uma frase ou palavra, servem para indicar que a palavra seguinte é um comando e não um dado ou texto.

Nos sistemas orientados por comandos, a "árvore" é muito superficial e abrangente, e uma única rotina, funcionando como programa de controle, guia o usuário até a sub-rotina requisitada. Esse "intérprete de comandos" tem quatro tarefas principais: 1) esperar por uma entrada do usuário; 2) analisar essa entrada — isto é, dividi-la em seus componentes funcionais; 3) interpretar o comando preparando o chamado à sub-rotina apropriada (Qual é o endereço da rotina? Há parâmetros que devem ser transferidos?); e, por fim, 4) chamar a rotina a ser executada. Quando o controle retorna, o intérprete retoma o início do ciclo.

O formato de um comando pode ser bastante complexo e alguns comandos são semelhantes à linguagem humana comum. Um exemplo de linguagem orientada por comandos é o sistema óperacional Unix, cujo formato típico de comando é:

Comando + lista opcional de parâmetros

Exemplo:

L

ou

L-1

Aqui, o comando L do Unix lista um diretório de arquivos, enquanto L-1 (onde -1 é um parâmetro opcional) lista um diretório de arquivos num formato "longo".

O analisador deve ser capaz de reconhecer as várias partes do comando. O Unix simplifica o procedimento, assumindo que a primeira palavra seja um comando e reconhecendo os parâmetros pelo sinal de menos que os precede. Embora inúteis para o intérprete de comandos, os parâmetros são exigidos pelas sub-rotinas por ele chamadas. As rotinas de um sistema orientado por comandos deveriam, idealmente, adotar um formato padronizado para os parâmetros de entrada. Se isso for feito, o intérprete de comandos transferirá os parâmetros na mesma forma em que foram introduzidos (como strings, por exemplo).

É muito mais fácil criar um intérprete de comandos do que desenvolver um sistema de menus. Usuários experientes tendem a preferir sistemas de comando, mais rápidos e mais flexíveis do que os programas orientados por menus. A maioria dos sistemas operacionais é orientada por comandos, o que representa uma infelicidade para os principiantes, uma vez que oferece poucas indicações e as rotinas de auxílio em tela (se existirem) exigem algum conhecimento do sistema. Além disso, o grande número de comandos e parâmetros opcionais em um típico sistema de comandos faz com que mesmo aqueles razoavelmente familiarizados com o sistema recorram ao auxílio na tela ou precisem consultar com freqüência o manual de operação.

Dois em três
Há, em geral, muitas maneiras de se solucionar o mesmo problema — estas telas mostram as diferentes opções de que um usuário dispõe para mudar as cores do fundo e das margens da tela, e também do próprio texto.

RADIOCONTATO

De origem modesta, a Motorola Incorporated é hoje uma das maiores empresas de componentes eletrônicos do mundo, com fábricas nos Estados Unidos e também na Europa.

Robert Galvin
Presidente da Motorola.

Como muitas outras empresas de sucesso, a Motorola começou com um único homem. Sua criação data de 1928, quando Paul Galvin fundou a Galvin Manufacturing Corporation, em Chicago, Estados Unidos. Ali ele se especializou na produção de rádios domésticos. Mas, nos anos 30, a empresa diversificou sua produção, fabricando rádios para a polícia e para automóveis, com a marca Motorola. Na década seguinte, a Motorola Incorporated — como já era conhecida — tornou-se uma das primeiras empresas eletrônicas a produzir semicondutores.

Paul Galvin faleceu em 1959 e seu filho Robert assumiu a presidência da empresa. Nos anos seguintes, outros fabricantes, sobretudo japoneses, começaram a competir com ela no mercado de semicondutores e da eletrônica em geral. A recessão mundial que teve lugar na metade da década de 70 provocou perdas vultosas para a Motorola, forçando-a a modificar sua estratégia. Ela contratou uma nova equipe, que incluía muitos técnicos de uma empresa rival, a Texas Instruments, e decidiu abandonar o campo da eletrônica tradicional — onde não podia mais competir —, concentrando-se na microeletrônica de alta tecnologia.

Isso implicava a venda de parte do patrimônio da Motorola (especialmente o setor de televisão em cores), o investimento de grandes somas em pesquisa e desenvolvimento e a compra de empresas em áreas novas, com o objetivo de causar impacto. O risco era considerável, mas havia pouca possibilidade de escolha.

Sede da Motorola,
em Illinois, EUA

No fim da década de 70, a Motorola estava bem longe das principais empresas do mercado de semicondutores, mas, depois de altos investimentos em tecnologia, passou a ameaçar a liderança da Texas Instruments, a maior do ramo.

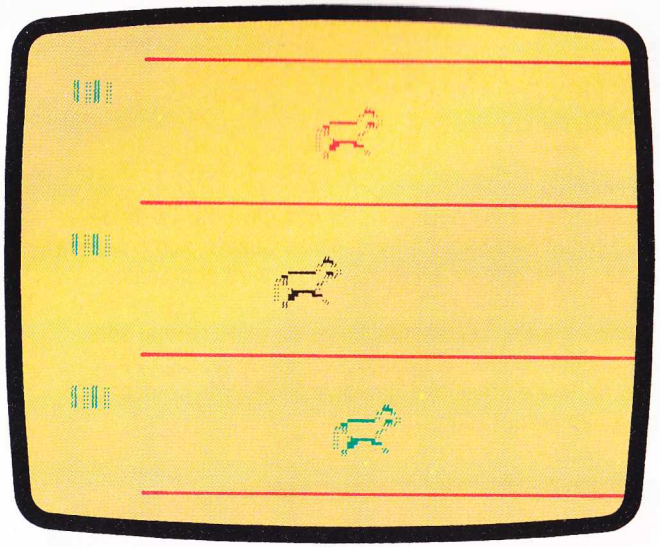
"As empresas concorrentes da Motorola já desapareceram porque não se adaptaram ao meio ambiente", comenta Robert Galvin.

A Motorola continuou, no entanto, a ter problemas para colocar seus produtos no mercado na época certa. Na metade da década de 70, quando a indústria do microcomputador dava os primeiros passos, o microprocessador de 8 bits 6800 da Motorola foi superado em vendas pelo chip Mostek 6502, principalmente por este ter sido adotado pela Apple para seus computadores pessoais de grande sucesso. Também o Intel 8085 e o Zilog Z80, usados em outros microcomputadores populares, impediram o êxito do 6800. Em 1976, a empresa lançou o 6809, reconhecidamente o melhor microprocessador de 8 bits. Mas a corrida pelo mercado já estava perdida e o chip só apareceu em poucos microcomputadores, como o Tandy e o Dragon.

Contudo, a empresa continuou a investir em pesquisa e em 1985 estava muito bem colocada na corrida pelo mercado de 16 bits. O microprocessador 68000 foi lançado em 1979, embora só se encontrasse distribuído no mercado em 1982. Adotaram-no os computadores Lisa e Macintosh, da Apple, e o QL, da Sinclair. Trata-se de um chip muito potente, que contém dezessete registros de 32 bits, um bus de dados de 16 bits e um bus de endereçamento de 24 bits.

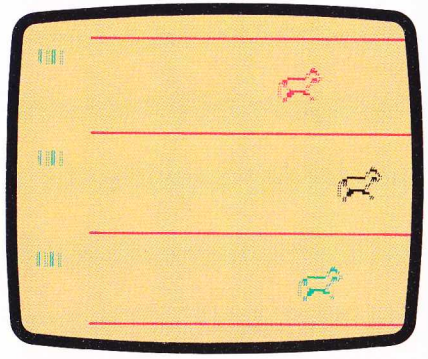
A Motorola continua a desenvolver novos produtos em seus centros de pesquisa em Phoenix, Arizona (EUA), em Genebra (Suíça) e em East Kilbride (Escócia). A fábrica de East Kilbride produz chips semicondutores complementares de óxido metálico (CMOS, Complementary Metal Oxide Semiconductors) e semicondutores de óxido metálico (MOS, Metal Oxide Semiconductors) para uma grande variedade de aplicações. A empresa está organizada em cinco grupos, trabalhando com comunicações, semicondutores, sistemas de informação, eletrônica automotiva e industrial e eletrônica militar.

Apesar de haver certa preocupação com a baixa lucratividade de alguns setores, a Motorola efetuou no primeiro trimestre de 1983 vendas no valor de 1,26 bilhão de dólares.



Para bom apostador
Tarde de Turfe é um típico jogo de apostas que permite mais de um participante. E o bom apostador pode mostrar sua competência no cálculo das probabilidades.

```
230 TEXT :HOME
240 PRINT "              VOCE ESTA ARRUINAD
    O!"
250 END
500 REM  ** ANDA UM PASSO  **
510 IF D(Q)=0 THEN HCOLOR=0:X=PEEK(-163
    36)
520 DRAW SH(Q) AT X(Q),Y(Q)
530 IF D(Q)=1 THEN 750
540 IF S(Q)=1 THEN S(Q)=0:GOTO 740
550 X(Q)=X(Q)+V(Q)
560 V(Q)=V(Q)+VP*RND(1)*(VM-V(Q))
570 IF S(Q)=2 THEN SH(Q)=3:Y(Q)=Y(Q)+12
    :S(Q)=0:GOTO 740
580 SH(Q)=SH(Q)+1:IF SH(Q)>3 THEN SH(Q)
    =2
590 IF (X(Q)>X1) AND (X(Q)<X3) THEN 710
600 IF X(Q)<XM THEN 740
610 X(Q)=X(Q)+20-XM
620 R(Q)=R(Q)+1
630 HCOLOR=3:HPLOT 3*R(Q),Y(Q)-20 TO 3*
    R(Q),Y(Q)-25
640 IF R(Q)<RD THEN 740
650 SH(Q)=1
660 IF WI=0 THEN WI=Q
670 D(Q)=1
680 FI=FI+1
690 X(Q)=20+8*(3-FI)
700 GOTO 740
710 IF X(Q)<X2 THEN SH(Q)=1:S(Q)=1:GOTO
    750
720 S(Q)=2:SH(Q)=2:Y(Q)=Y(Q)-12
730 GOTO 750
740 X=PEEK(-16336)
750 HCOLOR=3:DRAW SH(Q) AT X(Q),Y(Q)
760 RETURN
1000 REM  **  INICIALIZA CAVALOS **
1010 SH=1:WI=0:FI=0
1020 FOR K=1 TO 3
1030 X(K)=20:R(K)=0
1040 P(K)=FNRN(5)/5
1050 V(K)=P(K)*DV+V1
1060 Y(K)=K*40
1070 D(K)=0:SH(K)=1
1080 NEXT K
1090 REM  **  CHANCES E APOSTAS  **
1100 PT=0
1110 FOR I=1 TO 3
```



```
1120 P(I)=(V(I)-V1)/DV
1130 PT=PT+P(I)
1140 NEXT I
1150 FOR K=1 TO 3
1160 P(K)=INT(PT/P(K))
1170 NEXT K
1180 TEXT :HOME
1190 PRINT "**************************
     **********"
1200 PRINT "              PODE APOSTAR AGOR
     A"
1210 PRINT "**************************
     **********"
1220 PRINT
1230 PRINT "AS CHANCES SAO:"
1240 PRINT
1250 FOR I=1 TO 3
1260 PRINT "CAVALO ";I;" : PAGA ";P(I);"
      / 1"
1270 NEXT I
1280 PRINT
1290 FOR I=1 TO 3
1300 VTAB (11):HTAB (1):CALL -958
1310 PRINT "VOCE TEM AINDA ";CR;" MIL CR
     UZEIROS"
1320 PRINT
```



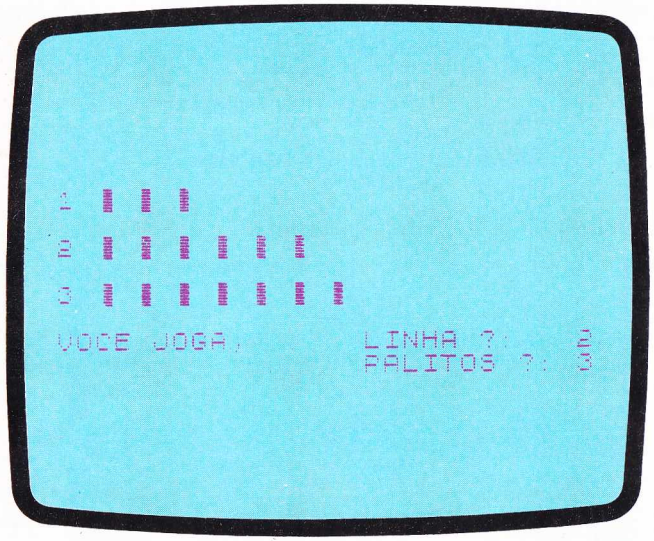
```
1330 PRINT "SUA APOSTA NO CAVALO "I" E':
     ";
1340 INPUT " ";I(I)
1360 RS=CR-I(I)
1370 IF RS>=0 THEN 1410
1380 PRINT
1390 PRINT "VOCE ANDA ACIMA DE SUAS POSS
     ES!"
1400 PRINT :GOTO 1330
1410 CR=CR-I(I)
1420 NEXT I
1430 RETURN
2000 REM  **  PREPARA O CAMPO     **
2010 HGR :HCOLOR=3
2020 HPLOT X(1),Y(1)-30 TO 279,Y(1)-30
2030 FOR K=1 TO 3
2040 HPLOT X(H),Y(K) TO X(H),Y(K)-8
2050 HPLOT X(K),Y(K)+10 TO 279,Y(K)+10
2060 DRAW SH(K) AT X(K),Y(K)
2070 NEXT K
2080 VTAB 21:HTAB 1:CALL -958
2090 PRINT "CAVALO:    1       2       3"
2100 PRINT "CHANCES:   ";P(1);TAB(18);P(2
     );TAB(25);P(3)
2110 PRINT "APOSTA:    ";I(1);TAB(18);I(2
     );TAB(25);I(3)
2120 PRINT " QUALQUER TECLA PARA COMECAR
      A CORRIDA";
2130 GET A$
2140 VTAB (24):HTAB (1):CALL -958
2150 PRINT CHR$(7);
2160 RETURN
5000 REM  **  INICIALIZA  **
5010 DEF FNRN(R)=INT(RND(1)*R+1)
5020 DIM R(3),V(3),X(3),Y(3),D(3),P(3),I
     (3),S(3),AD(3)
5030 V1=20:DV=10:VP=2:RD=10:XM=260:CR=10
     0
5040 XH=180:X1=XH-28:X2=XH-20:X3=XH+5
5050 AD(1)=775:AD(2)=815:AD(3)=856
5060 FOR SH=1 TO 3
5070 AD=AD(SH):GOSUB 5500:REM CORPO    DO
      CAVALO
5080 NEXT SH
5090 FOR SH=1 TO 3
5100 AD=AD(SH)+30
5110 READ A:REM PERNAS DO CAVALO
5120 POKE AD,A
5130 AD=AD+1
5140 IF A<>0 THEN 5110
5150 NEXT SH
5160 FOR I=1 TO 8
5170 READ A
5180 POKE 767+I,A
5190 NEXT I
5200 POKE 232,0:POKE 233,3:REM ENDERECO
     INICIAL DA TABELA DE FORMAS
5210 SCALE=1:ROT=0
5220 RETURN
5500 REM ** LE CORPO DO CAVALO  **
5510 RESTORE
5520 FOR I=1 TO 29
5530 READ A
5540 POKE AD+I,A
5550 NEXT I
5560 RETURN
6000 REM **  CORPO DO CAVALO     **
6010 DATA 36,36,12,12,101,45,45,45,101,1
     2,12,37,46,21
6020 DATA 14,246,28,55,30,30,30,30,63,63
     ,63,28,28,180,145
6030 REM ** PERNAS EM PE'        **
6040 DATA 45,30,55,110,73,73,36,60,44,4,
     0
6050 REM ** PERNAS CORRENDO 1    **
6060 DATA 45,30,191,110,72,72,72,45,21,2
     1,4,0
6070 REM ** PERNAS CORRENDO 2    **
6080 DATA 41,53,119,14,13,12,12,28,39,0
6090 REM ** FORMAS **
6100 DATA 3,0,8,0,48,0,89,0
```

PALITOS
PARA A LINHA SINCLAIR

Também chamado de Nim, o jogo dos palitos é muito antigo. No computador, que também faz as vezes do adversário, a disputa se inicia pela apresentação de três linhas, cada uma com número variável de palitos. Alternadamente, jogador e micro retiram certa quantidade de palitos de uma linha, perdendo quem tirar o último palito. Existem artifícios que permitem a um dos competidores comandar a partida no sentido de vencê-la. Aqui, a máquina emprega a melhor estratégia possível, mas não pode forçar a vitória. Isso cabe a você.

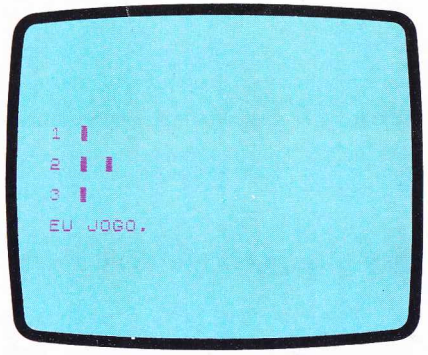
Com pequenas alterações nas linhas 280, 370 e 630, este programa também pode ser rodado no TK 90X.

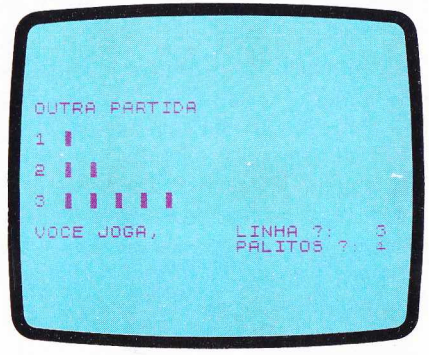
```
 10 REM
 20 REM        PALITOS
 30 REM  PARA A LINHA SINCLAIR
 40 REM
 50 LET P=1
 60 LET B=2
 70 LET J=3
 80 CLS
 90 PRINT "QUAL O SEU NOME?"
100 INPUT Z$
110 CLS
120 LET D=0
130 LET O=0
140 PRINT "OUTRA PARTIDA"
150 DIM A(J)
160 FOR E=P TO J
170 LET A(E)=INT (RND*7)+P
180 NEXT E
190 IF A(P)=A(B) OR A(B)=A(J) O
R A(P)=A(J) THEN GOTO 180
```



Vitória forçada
O segredo do jogo está em forçar a vitória, não ficando com o último palito. Por isso, há boas possibilidades de você vencer o computador.

```
 200 GOSUB 520
 210 IF E=5 THEN GOTO 160
 220 GOSUB 670
 230 PRINT "VOCE JOGA,","LINHA ?
 ";
 240 INPUT F
 250 PRINT F,,"PALITOS ?: ";
 260 INPUT H
 270 PRINT H
 280 PAUSE 100
 290 CLS
 300 IF F<P OR F>J OR H<P THEN G
OTO 570
 310 LET A(F)=A(F)-H
 320 IF A(F)<0 THEN GOTO 570
 330 IF A(1)+A(2)+A(3)=0 THEN GO
TO 580
 340 GOSUB 670
 350 PRINT "EU JOGO."
 360 LET Q=0
 370 PAUSE 100
 380 CLS
 390 FOR X=P TO 7
 400 FOR E=P TO J
 410 LET A(E)=A(E)-X
 420 IF A(E)<0 THEN GOTO 470
 430 IF Q=P THEN GOTO 220
 440 GOSUB 520
 450 IF E=5 THEN GOTO 220
 460 IF N=0 THEN GOTO 600
 470 LET A(E)=A(E)+X
 480 NEXT E
 490 NEXT X
 500 LET Q=P
 510 GOTO 390
 520 LET N=A(P)+A(B)+A(J)
 530 LET N=N+(INT (A(P)/2)+INT (
A(B)/2)+INT (A(J)/2))*8
 540 LET N=N+(INT (A(P)/4)+INT (
A(B)/4)+INT (A(J)/4))*80
 550 IF (N=222)+(N=220)+(N=202)+
(N=200)+(N=22)+(N=20)+(N=J)+(N=P
)=1 THEN LET E=5
 560 RETURN
 570 PRINT "--LANCE INVALIDO--",
,;
 580 LET D=D+P
 590 GOTO 610
 600 LET C=C+P
 610 PRINT "PONTOS"
 620 PRINT Z$," : ";C
 630 PAUSE 180
 640 IF D=5 OR C=5 THEN GOTO 60
 650 CLS
 660 GOTO 140
 670 PRINT
```





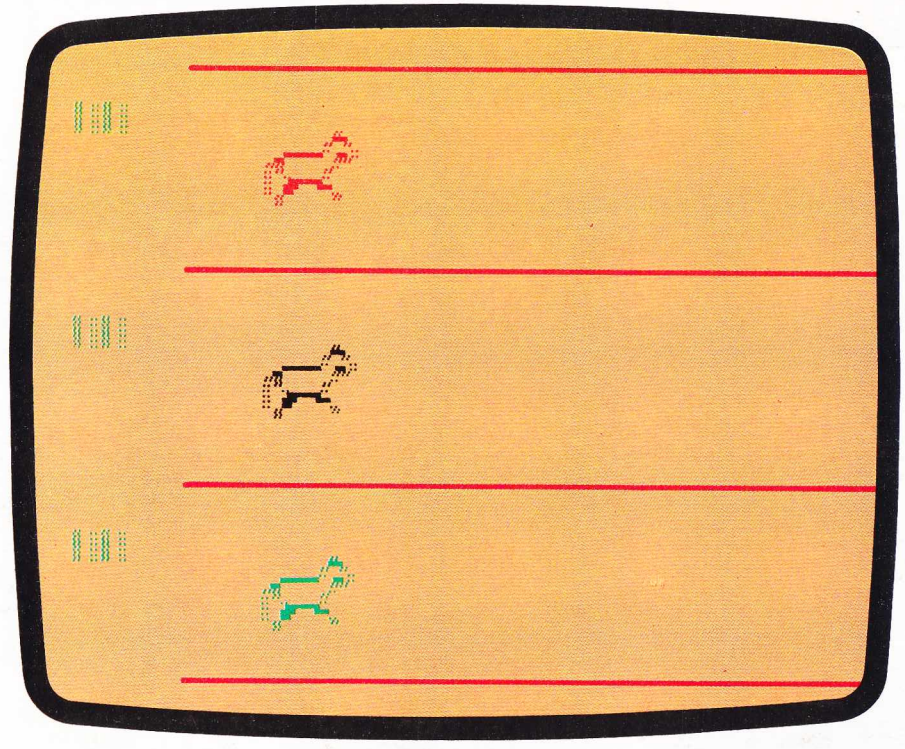

```
 680 FOR E=P TO J
 690 PRINT E;" ";
 700 IF A(E)=0 THEN GOTO 740
 710 FOR X=P TO A(E)
 720 PRINT "■ ";
 730 NEXT X
 740 PRINT
 750 PRINT
 760 NEXT E
 770 RETURN
```


TARDE DE TURFE

PARA LINHA APPLE

Cuidado com seu dinheiro, pois agora você vai apostar nos cavalos. Tarde de Turfe traz o hipódromo ao computador. Sem tantos cavalos quanto numa corrida real, é verdade — apenas três alinham para a largada. Mas, para compensar, eles dão dez voltas pela raia.

O jogo se desenvolve como no turfe: as possibilidades de cada animal são indicadas no início, e você poderá apostar enquanto tiver dinheiro. Siga as instruções de aposta dadas na tela e pressione qualquer tecla para começar a corrida. Seja moderado para não perder tudo.



```
 10 REM ****************************
 11 REM *      PARA LINHA APPLE     *
 12 REM *                           *
 13 REM *       TARDE DE TURFE      *
 15 REM ****************************
 20 GOSUB 5000
100 REM  ** LOOP PRINCIPAL **
110 GOSUB 1000
120 GOSUB 2000
130 VM=(V(1)+V(2)+V(3))/3
```

```
140 FOR Q=1 TO 3
150 GOSUB 500
160 NEXT Q
170 IF FI<3 THEN 140
180 INVERSE :PRINT "       O CAVALO ";WI
    ;" E' O VENCEDOR !!!       ";
190 NORMAL
200 CR=CR+I(WI)*P(WI)
210 FOR I=1 TO 1500:W=I/I+1:NEXT
220 IF CR>0 THEN 100
```